



DMM Technology Corp.

DYN AC Servo Drive CAN Specification

[DYNCAN1-BL314-16A]

Document Version 1.6A
Published Feb 12, 2026

1. Overview

The DYN servo drive follow standard CAN2.0A specification. The following DYN servo drive model numbers supports this CAN protocol:

- | | |
|---------------|---------------|
| DYN2-T1C6S-00 | DYN4-L01C2-00 |
| DYN2-TLC6S-00 | DYN4-H01C2-00 |
| | DYN4-T01C2-00 |

Baud Rate and Frame Format:

Baud Rate (kHz)	Data Frame
<ul style="list-style-type: none"> 10, 20, 50, 125, 250, 500, 800, 1000 	<ul style="list-style-type: none"> Base frame format with 11 identifier bits 1~4 bytes data 11-bit identifier includes both drive ID and function code

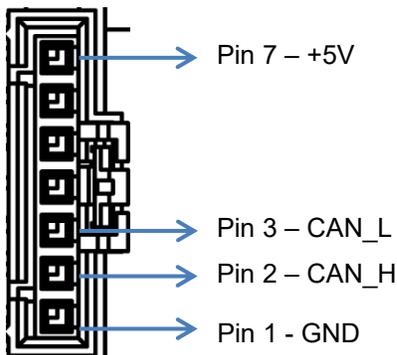
The DYN servo drive RS232 port is still active when in CAN mode. The user can continue to use the RS232 port to communicate with the DMMDRV program while in CAN mode.

All servo drive parameters, excluding Max Speed and Max Acceleration, limited to 1,000,000read/write cycles to protect EEPROM memory.

2. Hardware Interface

Connector

Both the DYN2 and DYN4 servo drive shares the same connection method.



Port: JP2
 Connector Type: 2.54mm Pitch Rectangular
 Drive Header: (Molex) 70553-0041
 Plug Connector: (Molex) 50-57-9407

Use twisted pair cables for the CANH and CANL lines. Make sure ground is connected between CAN controller and servo drive.

Recommended CAN adapter for Development:

Manufacturer	PEAK Systems
Model#	PCAN-USB
Website	https://www.peak-system.com/PCAN-USB.199.0.html?&L=1
Supplier	http://www.phytools.com/PCAN_USB_Adapter_p/ipeh-002021.htm

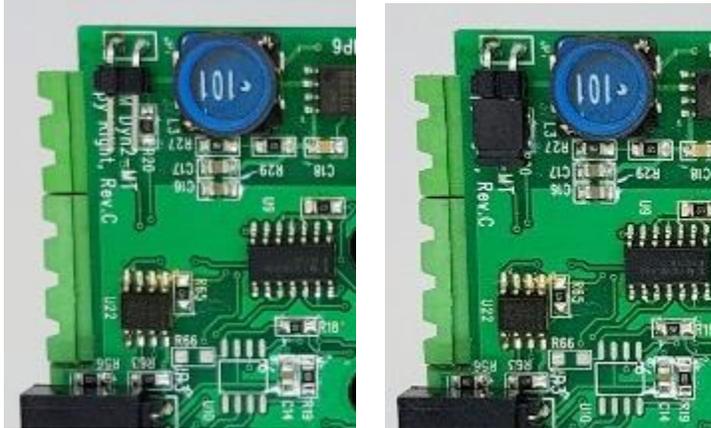
Terminating Resistor

Both the DYN2 and DYN4 servo drive can be optioned with terminating resistors (200ohm) which can be selected via jumpers on the servo drive PCB. Terminating resistors can also be connected externally on the signal lines.

Terminating Resistor Installation Instructions

DYN2 Servo Drive:

1. Power OFF servo drive and wait 5 minutes for drive to fully discharge.
2. Remove servo drive cover and locate terminating resistor jumper.



3. Inserting the jumper connects the terminating resistor. Removing the jumper disconnects the terminating resistor. By factory default, the servo drive does not have the jumper connected.

DYN4 Servo Drive:

1. Power OFF servo drive and wait 5 minutes for drive to fully discharge.
2. Remove front cover of servo drive and locate terminating resistor jumper.



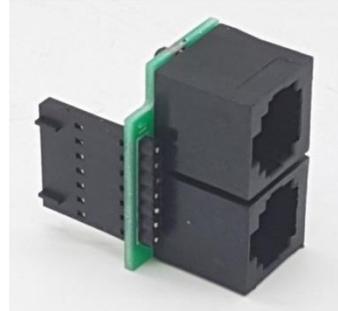
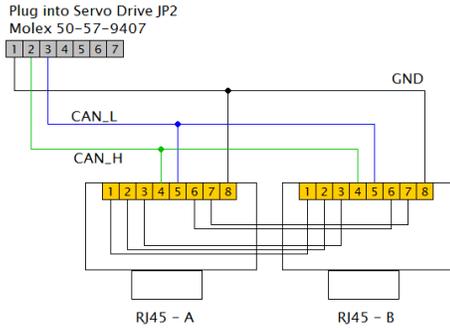
3. Inserting the jumper connects the terminating resistor. Removing the jumper disconnects the terminating resistor. By factory default, the servo drive does not have the jumper connected.

JP2 RJ45 Splitter

The RJ45 splitter can also be used to network multiple drives using standard RJ45 modular cables and connectors. The splitter connects into either DYN2 or DYN4 JP2 ports and splits the CAN_H, CAN_L and Ground signals between two RJ45 ports. The RJ45 splitter is sold separately.

Part# CNJP2-RJ45SP-2

Splitter Connector Circuit Diagram:



DYN2 and DYN4 servo drives networked with RJ45 Splitter Connector:

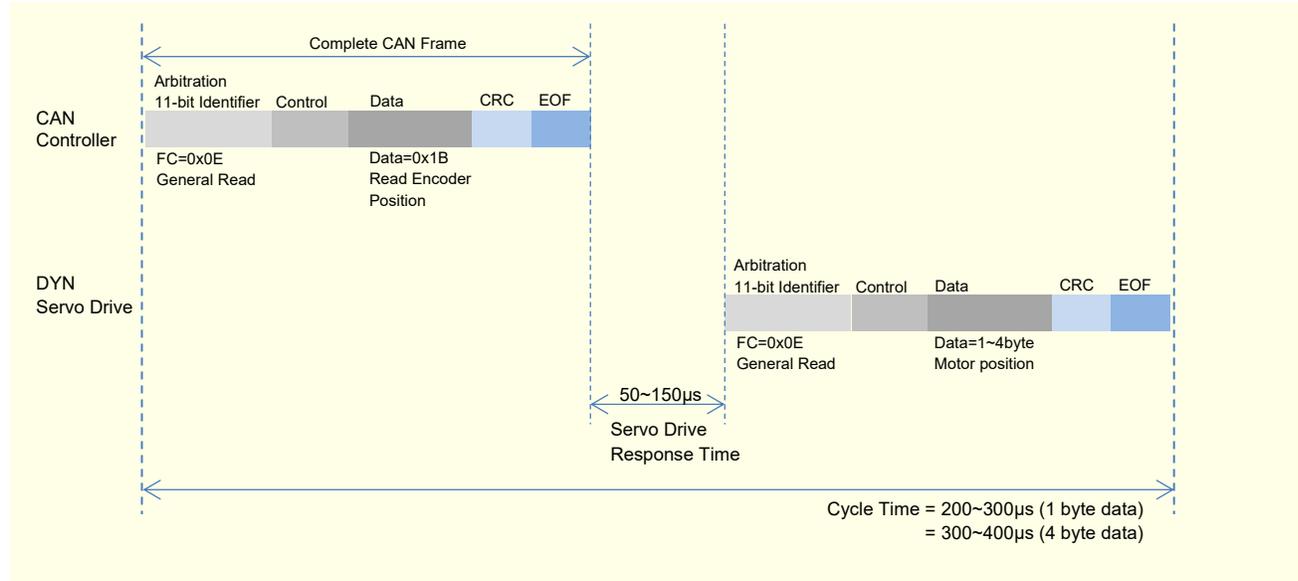


When using the RJ45 Splitter Connector, the RJ45 terminating resistor can be used Part# CNJP2-RJ45-TM1:



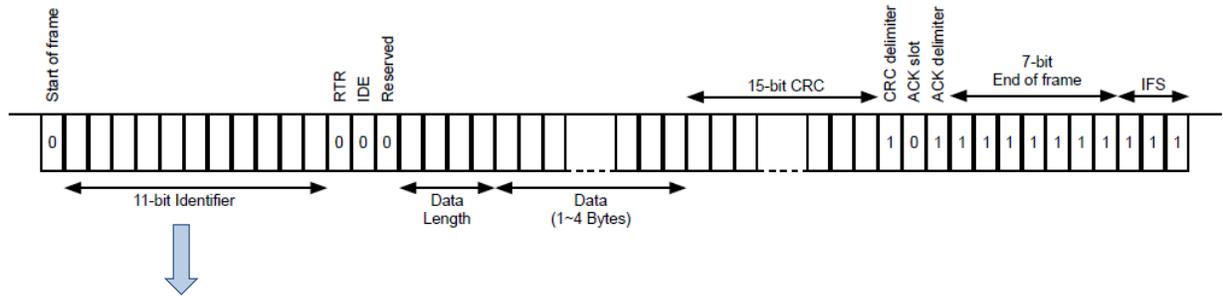
3. Servo Drive message response time / Cycle Time

Typical servo drive response time to read encoder position. 500kHz baud. See section 4 for message details.



4. CAN Frame Details

DYN servo drive CAN Protocol Data Framing:



11-bit Identifier Structure:

Drive ID bit 5	Drive ID bit 4	Drive ID bit 3	Drive ID bit 2	Drive ID bit 1	Drive ID bit 0	FC bit 4	FC bit 3	FC bit 2	FC bit 1	FC bit 0
b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

11-bit Identifier Consists both Drive ID and Command Function Code:

b4~b0 = 5-bit Function Code (FC)

b5~b10 = Drive ID 0~64

RTR Field:

Remote transmission request (RTR): The DYN servo drive CAN protocol does not support Remote Frame functionality.

Data Length and Data Field:

Data Length Bytes	Data Range
1	-128 ~ 127
2	-32,768 ~ 32,767
3	-8,388,608 ~ 8,388,607
4	-134,217,728 ~ 134,217,727

5. Function Code Outline

	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)	Details
0	Set_Origin	0x00	1	1	N/A	0	Set absolute zero origin
1	Go_Absolute_Pos_PTP	0x01	1~4	1~4	-134,217,728 ~ 134,217,727	0	Point to Point Absolute Move
2	N/A						
3	Go_Relative_Pos_PTP	0x03	1~4	1~4	-134,217,728 ~ 134,217,727	0	Point to Point Relative Move
4	N/A						
5	Assign_Drive_ID	0x05	1	1	0~64	0	Assign Drive ID
6	Read_Drive_ID	0x06	0	0	0~64	1	Read Drive ID
7	Set_Drive_Config	0x07	1	1	N/A	0	Set Drive configuration
8	Read_Drive_Config	0x08	1	0	N/A	1	Read Drive configuration
9	Read_Drive_Status	0x09	0	0	N/A	1	Read Drive status
10	Turn_ConstSpeed	0x0a	1~2	1~2	-20,000~20,000	0	Turn constant speed command
11	Square_Wave	0x0b	1~3	1~3	0~4096	0	Square wave motion command
12	Sin_Wave	0x0c	1~3	1~3	0~4096	0	Sine wave motion command
13	SS_Frequency	0x0d	1	1	0~50	0	Square/Sine wave motion frequency
14	General_Read	0x0e	1~4	1	-134,217,728 ~ 134,217,727	1~4	Read motor position or torque
15	ForMotorDefine	0x0f	N/A	N/A	N/A	N/A	Internal Function - Not user accessible
16	Set_Param_Group1	0x10	1	4	N/A	0	Set_MainGain, Speed Gain, Int Gain, Torque Filter Constant
17	Set_Param_Group2	0x11	1	2	N/A	0	Set High Speed, High Accel
18	Set_Param_Group3	0x12	1	5	N/A	0	Set_ON_POS_RANGE, GEAR_NUM, LINE_NUM
19	Read_Param_Group1	0x13	1	0	N/A	4	Read Main Gain, Speed Gain, Int Gain, Torque Filter Constant
20	Read_Param_Group2	0x14	1	0	N/A	2	Read High Speed, High Accel
21	Read_Param_Group3	0x15	2	0	N/A	6	Read ON_POS_RANGE, GEAR_NUM, LINE_NUM
22	Go_Absolute_Pos_Sync	0x16	1~4	1~4	-134,217,728 ~ 134,217,727	0	Synchronized ABS move command
23	Go_Relative_Pos_Sync	0x17	1~4	1~4	-134,217,728 ~ 134,217,727	0	Synchronized REL move command
24	Sync_Trigger	0x18	1	1	N/A	0	Synchronized move command trigger
25	Servo_Disable	0x19	1	1	N/A	0	Servo Enable/Disable Input
26	Read_Motor_Speed	0x1A	2	0	N/A	2	Read motor speed
27	Drive_Reset	0x1B	1	1	N/A	1	Servo Drive Reset Command
28	Go_Relative_Pos_PD	0x1C	1~2	1~2	-16384~16384	0	Position Direct Relative Move
29	Drive_Error	0x1D	1	0	N/A	1	CAN error message output
30	Echo_Function	0x1E	4	4	N/A	4	4 byte data echo
31	Diagnostic_counter	0x1F	1	0	N/A	1	Diagnostic counter output

N/A = Not Applicable

6. CAN Function Details

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
0	Set_Origin	0x00	0	0	N/A	0

Details

Sending data 0xAB sets the current motor position as the absolute zero position.

When the drive is set to operate in Absolute Mode (Configuration 0x04=1), when the drive powers ON, the motor moves to the set absolute zero position, then starts accepting command.

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
1	Go_Absolute_Pos_PTP	0x01	1~4	1~4	-134,217,728 ~ 134,217,727	0

Details

Point to Point Move Absolute Position command. Positive command turns motor in CW direction, negative command turns motor in CCW direction. Drive runs motion as soon as command received.

Point to Point command acceleration/deceleration determined by Max Acceleration parameter. Speed determined by Max Speed Parameter.

Servo drive return CAN Data Error if command position outside allowed data range.

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
3	Go_Relative_Pos_PTP	0x03	1~4	1~4	-134,217,728 ~ 134,217,727	0

Details

Point to Point Move Relative Position command. Positive command turns motor in CW direction, negative command turns motor in CCW direction. Drive runs motion as soon as command received.

Point to Point command acceleration/deceleration determined by Max Acceleration parameter. Speed determined by Max Speed Parameter.

Servo drive return CAN Data Error if command position outside allowed data range.

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
5	Assign Drive ID	0x05	1	1	N/A	0

Details

Assigns drive ID in servo. Requires power cycle after changing to re-establish communication.

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
6	Read Drive ID	0x06	0	0	0~64	1
Details						
Reads 1 byte servo drive ID number.						

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
7	Set_Drive_Config	0x07	1	1	N/A	0
8	Read_Drive_Config	0x08	1	0	N/A	1
Details						
Set and read servo drive configuration byte.						
Drive Configuration = b7 b6 b5 b4 b3 b2 b1 b0						
<p>Command Input Mode</p> <p>b1 b0 = 0 = RS232 mode 1 = CW,CCW mode 2 = Pulse/Direction mode 3 = Analog mode</p> <p>Relative or Absolute Mode (Encoder Mode)</p> <p>b2 = 0 = Works as relative mode. Operates as incremental encoder. 1 = Works as absolute mode. At power up, motors moves to absolute zero or stored zero position. See <i>Set ABS Origin</i> command for details.</p> <p>Servo Mode</p> <p>b4 b3 = 0 = Position Servo Mode (default for Modbus) 1 = Speed Servo Mode 2 = Torque Servo Mode</p> <p>Servo Enable/Disable</p> <p>b5 = 0 = Servo Enabled 1 = Servo Disabled (Motor Free)</p> <p>b7 b6 = Unimplemented</p>						

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
9	Read_Drive_Status	0x09	0	0	N/A	1
Details						
<p>Read the servo drive Status.</p> <p>Drive Status = b7 b6 b5 b4 b3 b2 b1 b0</p> <p>b0 = 0 = On position. Pset - Pmotor <= OnpositionRange 1 = Off Position / motor busy. Pset - Pmotor > OnPositionRange</p> <p>b1 = 0 = Servo Enabled 1 = Servo Disabled / Motor Free</p> <p>b4 b3 b2 = 0 = No Alarm 1 = Motor lost phase alarm, Pset - Pmotor >8192(steps), 180(deg) 2 = Over current alarm 3 = Overheat alarm / Overpower alarm 4 = Error for CRC code check, refuse to accept current command</p> <p>b5 = 0 = Built in S-curve, linear, circular motion completed; waiting for next motion 1 = Built in S-curve, linear, circular motion is busy on current motion</p> <p>b7 b6 = Unimplemented</p>						

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
10	Turn_ConstSpeed	0x0a	1~2	1~2	-20,000~20,000	0
Details						
<p>Constant speed command function. Sending command immediately starts motor rotation, servo drive internally automatically switch into speed mode. Max Acceleration parameter controls acceleration when changing speed. Command is in rpm units. Positive command turns motor in CW direction, negative command turns motor in CCW direction. Servo drive return CAN Data Error if speed command is not within -20,000~20,000.</p> <p>Example:</p> <ul style="list-style-type: none"> - To send speed command of 50rpm, send 1 data byte B0=0x32 - To send speed command of -2000rpm (0xF830), send 2 data byte B0=F8 B1=30 						

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
11	Square_Wave	0x0b	1~3	1~3	0~4096	0
Details						
<p>Sets built-in Square Wave motion amplitude. 1024 = 45degree, 2048 = 90degree, 4096 = 180degree amplitude.</p> <p>Motion begins when register is command is given, and SS_Frequency command is not zero. Servo drive return CAN Data Error if sent data not within range.</p>						

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
12	Sin_Wave	0x0c	1~3	1~3	0~4096	0
Details						
Sets built-in Sine Wave motion amplitude. 1024 = 45degree, 2048 = 90degree, 4096 = 180degree amplitude. Motion begins when register is set, and SS_Frequency command is not zero. Servo drive return CAN Data Error if sent data not within range.						

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
13	SS_Frequency	0x0d	1	1	0~50	0
Details						
Sets built-in Square/Sine Wave motion frequency. Units in Hertz. Setting resets to zero when drive powered OFF. Servo drive return CAN Data Error if sent data not within range.						

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
14	General_Read	0x0e	1~4	1	-134,217,728 ~ 134,217,727	1~4
Details						
Read 32-bit motor position or torque command. To read motor 32-bit absolute encoder position, send 0x0e function code with 1 byte data 0x1b . Drive will respond with 32-bit signed absolute encoder position. Data length depends on position decimal value – see Data Length and Data Field in section 3. Absolute position is positive in CW direction, negative in CCW direction. To read motor torque, send 0x03 function code with 1 byte data 0x1e . Drive will respond with 1~2byte torque value. Torque value range is [-981:+981]. 981=peak output current of servo drive. Value is positive when current/torque is applied in CW direction. Example: <ul style="list-style-type: none"> - Servo drive used = DYN4-H01C2. Peak output current = 20A - Servo motor used = 11A-DHT-A6HK1. Torque coefficient = 0.774Nm/A - Motor Torque read value = 0xFF63 = -157 - $157 / 981 = 0.224 * 20A = 3.2A$ - $3.2A * 0.774Nm/A = 2.48Nm$ applied in CCW direction since reading is negative 						

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
15	ForMotorDefine	0x0f	N/A	N/A	N/A	N/A
Details						
Proprietary DMM functionality. Not user accessible. Do not write to this function code.						

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
16	Set_Param_Group1	0x10	1	4	N/A	0
Details						
<p>Sets Parameter Group1 including Main Gain, Speed Gain, Integration Gain and Torque Filter Constant.</p> <p>Send 0x10 function code with 4 byte data. First byte Main Gain, second byte Speed Gain, third byte Integration Gain, fourth byte Torque Filter Constant.</p> <p>Sending 0x00 data byte does not save that parameter. All parameter has setting range [1:127]. Servo drive return CAN Data Error if sent data not within range.</p>						

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
17	Set_Param_Group2	0x11	1	2	N/A	0
Details						
<p>Sets Parameter Group 2 including Max Speed, Max Acceleration.</p> <p>Max Speed and Max Acceleration saves into EEPROM first 10 times after power ON. After saves into RAM and is erased when drive powered OFF.</p> <p>Send 0x11 function code with 2 byte data. First byte Max Speed, second byte Max Acceleration.</p> <p>Sending 0x00 data byte does not save that parameter. All parameter has setting range [1:127]. Servo drive return CAN Data Error if sent data not within range.</p>						

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
18	Set_Param_Group3	0x12	1	5	N/A	0
Details						
<p>Sets Parameter Group1 including On Pos Range, GEAR_NUMBER (2 bytes), and LINE_NUMBER (2 bytes).</p> <p>Send 0x12 function code with 5 byte data. First byte On Pos Range, second and third byte is GEAR_NUMER, fourth and fifth byte is LINE_NUMER.</p> <p>Sending 0x00 data byte does not save that parameter. On Pos Range parameter has setting range [1:127]. GEAR_NUMBER has setting range [500:16384]. LINE_NUMER has setting range [500:4095].</p> <p>Servo drive return CAN Data Error if sent data not within range.</p>						

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
19	Read_Param_Group1	0x13	1	0	N/A	4
Details						
Reads Parameter Group1. Servo drive returns 4 bytes. First byte Main Gain, second byte Speed Gain, third byte Integration Gain, fourth byte Torque Filter Constant.						

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
20	Read_Param_Group2	0x14	1	0	N/A	2
Details						
Reads Parameter Group2. Servo drive returns 2 bytes. First byte Max Speed, second byte Max Acceleration.						

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
21	Read_Param_Group3	0x15	2	0	N/A	6
Details						
Reads Parameter Group3. Servo drive returns 6 bytes with little endian format. First two bytes ON_POS_RANGE, second and third byte is GEAR_NUMER, fourth and fifth byte is LINE_NUMER.						

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
22	Go_Absolute_Pos_Syn c	0x16	1~4	1~4	-134,217,728 ~ 134,217,727	0
Details						
Synchronized PTP Move Absolute Position command. Sending data loads Synchronized Absolute Position command. Servo drive does not run motor. Motion begins when Sync_Trigger (FC=0x18) and 0xFF data command received. Servo drive return CAN Data Error if sent data not within range.						

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
23	Go_Relative_Pos_Sync	0x17	1~4	1~4	-134,217,728 ~ 134,217,727	0
Details						
Synchronized PTP Move Relative Position command. Sending data loads Synchronized Relative Position command. Servo drive does not run motor. Motion begins when Sync_Trigger (FC=0x18)) and 0xEE data command received. Servo drive return CAN Data Error if sent data not within range.						

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
24	Sync_Trigger	0x18	1	1	N/A	0

Details

Synchronized PTP Move start motion trigger.

Sending 1 byte data 0xFF starts Go_Absolute_Pos_Sync motion with loaded position.

Sending 1 byte data 0xEE starts Go_Relative_Pos_Sync motion with loaded position.

For multi-axis applications, using Sync motion command allows better synchronization between each axis. First, load all axis with command. Then send all Sync_Trigger command to start motion. Use broadcast ID=0 to send Sync Trigger command to multiple axis simultaneously.

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
25	Servo_Disable	0x19	1	1	N/A	0

Details

Servo enable/disable command. Send 1 byte data 0x00 to enable servo. Send 1 byte data 0xFF to disable servo.

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
26	Read_Motor_Speed	0x1A	2	0	N/A	2

Details

Servo drive responds with current motor speed. Units in [rpm].

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
27	Drive_Reset	0x1B	1	1	N/A	1

Details

Servo Drive Reset Command.

Send 1 byte data 0xFE to reset servo drive. Reset clears fault and resets motor absolute position.

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
28	Go_Relative_Pos_PD	0x1C	1~2	1~2	-16384~16384	0

Details

Send move relative position command directly to servo without PTP calculation. Servo drive runs command under max acceleration and speed. Position Direct (PD) command.

Command is used for interpolation, CAM, and profile following commands where controller sends continuous position command to servo drive.

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
29	Drive Error	0x1D	1	0	N/A	1
Details						
<p>CAN error message and servo drive Alarm output.</p> <p>If controller sends CAN message with invalid data range, servo drive responds with fc=0x1D and data = 0xFF to indicate data error.</p> <p>When servo drive Alarms, servo drive instantaneously sends CAN message with FC=0x1D and data = 0xEE to indicate alarmed state. Controller should monitor and stop operation when this message is received.</p>						

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
30	Echo_Function	0x1E	4	4	N/A	4
Details						
<p>Echo function. Controller sends 4 byte data with FC=0x1E, Servo drive responds with same 4 byte data.</p>						

#	Command	5-bit FC	Total Data (Bytes)	Sent Data (Bytes)	Data Range (Decimal)	Return Data (Bytes)
31	Diagnostic_counter	0x1F	1	0	N/A	1
Details						
<p>Drive internal unsigned 8-bit counter used for testing and diagnostics. Servo drive sends counter number when FC=0x1F command sent. Returned value increments by 1 each time it is read. Rolls back to 0 after 255.</p> <p>Can be used for CAN network management or heartbeat functionality.</p>						

All specified data subject to change without notice to reflect updates and improvements made to product. DMM Technology Corp. warrants the quality and performance of for one year starting date of shipment from original factory. DMM Technology Corp. assumes no responsibility for damages resulting from user related errors or improper use of product, in which case the warranty terms will be void. Safety precautions should be considered for all applications. As this product does not include safety conditions, always design a higher-level feedback to reduce the risks of product or bodily harm.

DMM TECHNOLOGY CORP.

120 - 21320 Gordon Way Richmond, British Columbia V6W1J8 Canada

PHONE: +1 (604)-370-4168 | FAX: +1 (604) 285-1989

WEB: <http://www.dmm-tech.com>

SALES: sales@dmm-tech.com

INFO: info@dmm-tech.com