Applied Motion Products CANopen Manual

APPLIED MOTION PRODUCTS, INC.

920-0025 Rev. G
12/14/2010

(This page intentionally left blank)

Introduction

This manual describes Applied Motion Products CANopen implementation of CiA DS-301 and CiA DSP-402 specifications. The reader is expected to fully understand both CiA standards and along with this specification, will be able to develop a distributed motion control system.

The intent of this manual is to highlight manufacturer specific requirements as they pertain to Applied Motion Products drives.

Information and standards may be obtained from the CiA website at http://www.can-cia.com/. Information and software relating directly to the Applied Motion Products drives, including an open-source example program, may be obtained from our website at http://www.applied-motion.com/support.

Contents

List of Tables

List of Figures

(This page intentionally left blank)

Reference Documents

ST5/10-C EDS
SV7-C EDS
STAC6 EDS
CiA DS-301
CiA DS-303
CiA DSP-402
Bosch CAN Physical Layer Specifications 2.0B
Applied Motion Products Q Command Reference

Commonly Used Acronyms

CiA	CAN in Automation Group (Standards Body)
CAN	Controller Area Network
COB ID	Communication Object Identification
DR	CiA Draft Recommendation
DS	CiA Draft Standard
DSP	CiA Draft Standard Proposal
EDS	Electronic Data Sheet
NMT	Network Management
OD	Object Dictionary
PDS	Power Drive System
PDO	Process Data Object
RPDO	Receive (incoming) PDO
SDO	Service Data Object
TPDO	Transmit (outgoing) PDO

CANopen Network Topology Overview

Applied Motion Products CANopen drives can be integrated into a CANopen system with other device types as shown below.

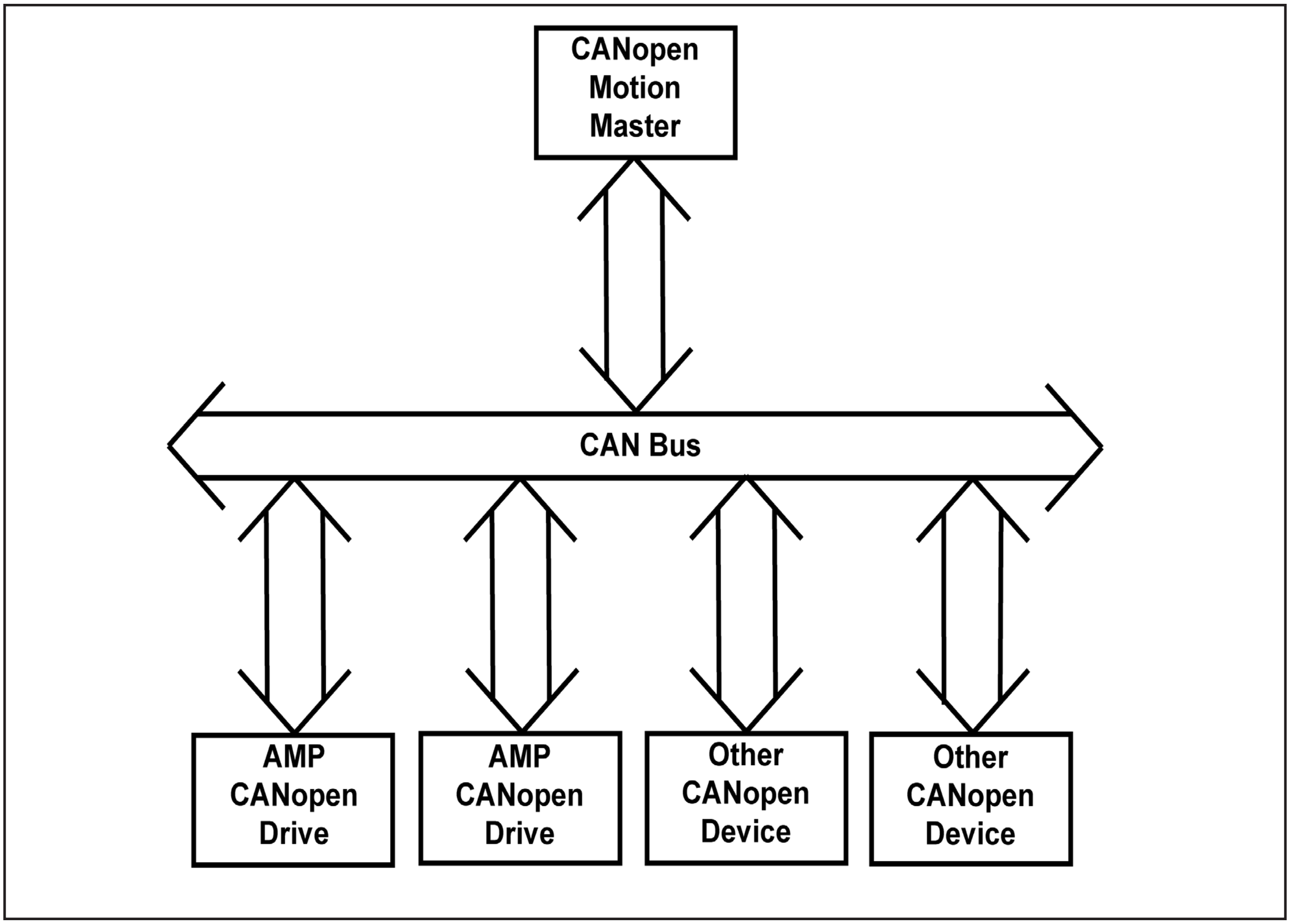


Figure 1: CANopen Network Topology Overview

Drive Setup

There are four phases to setting up an AMP CANopen Drive:

- Wiring the power and motor
- Wiring the CANopen connector to the drive
- Setting the Bit Rate and Node ID
- Configuring the Drive

Wiring the Power and Motor

Please refer to your drive's hardware manual for this step. The appropriate manual may be downloaded at http://www.applied-motion.com/support/manuals.

Wiring the CANopen Connector for ST5-C Drives

As an example, the Applied Motion Products ST5-C drive uses a four-pin spring connector, shown in Figure 2 below, and conforms to DR303 specification. The connector should be wired in a daisy-chain configuration, as shown in Figure 3 below, with a 120 ohm resistor used to terminate each end. Other wiring topologies, such as star networks, are not recommended due to wave reflection problems. Please reference specific hardware manuals for your drive's wiring configuration.

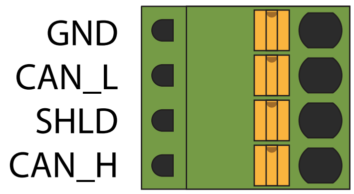
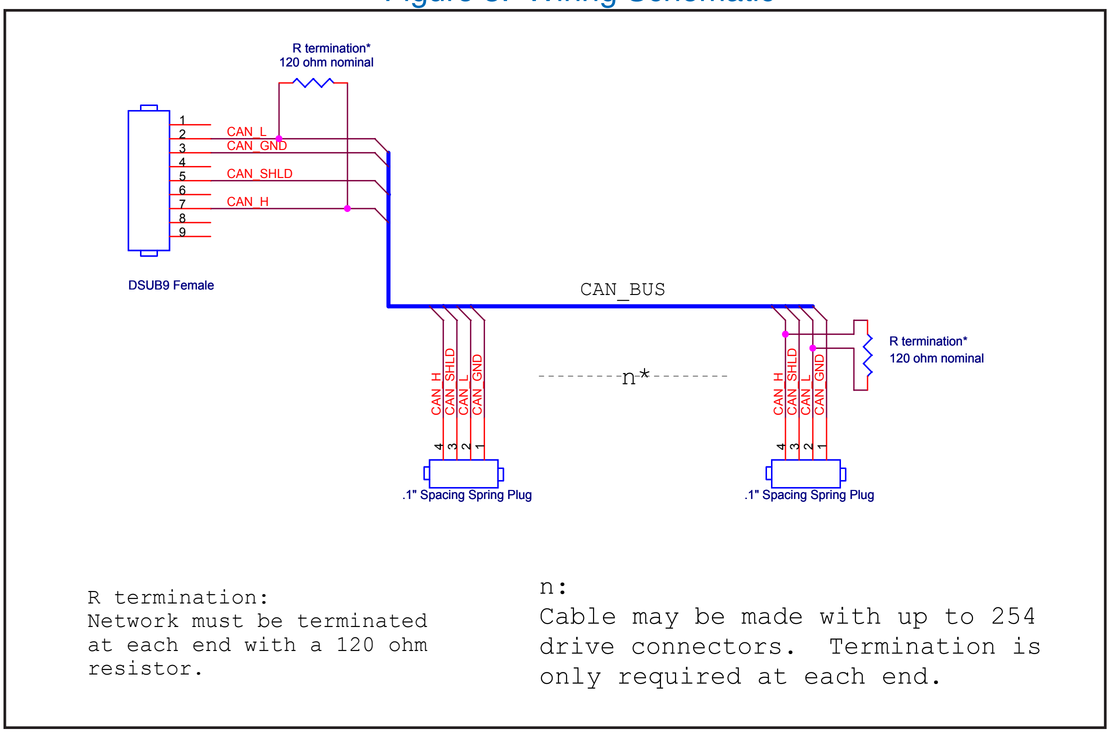


Figure 2: The CANopen Connector

Figure 3 shows a CANopen network with two AMP ST5-C drive connectors wired to a nine-pin D-sub style CAN connector.

Figure 3: Wiring Schematic



Example, Wiring the CANopen Connector for ST5-C Drives to a Kvaser Leaf USB to CANopen Adapter

CANopen Bitrate

AMP CANopen drives have three settings, one for Bit Rate and two for Node ID.

- The Bit Rate is configured using an eight-position switch. See Table 1 for the Bit Rate settings. Please reference your drive's hardware manual for the location of the Bit Rate switch.
- The Node ID is configured using a sixteen position switch to set up the lower four bits of the Node ID and a seven position switch to set up the upper three bits of the Node ID. In some cases, the upper three bits of the Node ID are configured by using the Configurator or QuickTuner. Please reference your drive's hardware manual for Node ID switch configuration and setup. Valid ranges for the Node ID are 0x01 through 0x7F. Node ID 0x00 is reserved in accordance to DS 301 specification.

Note: The Node ID and Bit Rate is captured only after a power cycle, or after a network reset command has been sent. Changing the switches while the drive is powered on will NOT change the Node ID until one of those conditions has also been met.

Switch Setting	Resultant Bit Rate
0	1 Mbps
1	800kbps
2	500 kbps
3	250 kbps
4	125 kbps
5	50 kbps
6	20 kbps
7	12.5 kbps

Table 1: Bit Rate Switch Settings

Drive Configuration

Once the CAN connector has been wired to the drive, and the Node ID and Bit Rate have been set, it's time to configure the drive. Drive configuration for ST stepper drives and STM drive+motors is accomplished using our *ST Configurator* software, which can be found by visiting http://www.applied-motion.com/products/software. To configure a STAC6, use *STAC Configurator*. Drive configuration and tuning for servo drives is performed using *Quick Tuner*. In all cases you'll need to connect the drive to a Windows PC using the included RS-232 serial cable. Please refer to the appropriate software manual or built-in help screens for details.

Note: When the CANopen drive is first powered on, the drive will power up and automatically sends a power-up packet over the RS232 port. If an Applied Motion Products application is present, it will send a response back to the drive over RS232 and the drive will hold the CAN node in the "Initialization" state until the application is closed. If no response is detected, the drive continues the normal CANopen start-up procedure, that is, the drive will power up into the "Initialization" state, send out a boot-up packet, and move into the "Pre-Operational" state and start sending out heartbeats with the "Pre-Operational" state as a status code.

Supported DSP402 Modes of Operation

	ST	STM	STAC6	SV
Profile Velocity	•	•	•	•
Profile Position	•	•	•	•
Profile Torque				•
Homing	•	•	•	•
Interpolated Position				
Q Program	•	•	•	•

Table 2: Modes of Operation

For detailed information:

- **Profile Position Mode - See Appendix C**
- **Profile Velocity Mode - See Appendix D**
- **Homing Modes - See Appendix E**
- **Profile Torque Mode - See Appendix F**
- **Q Program Mode - See Appendix G**

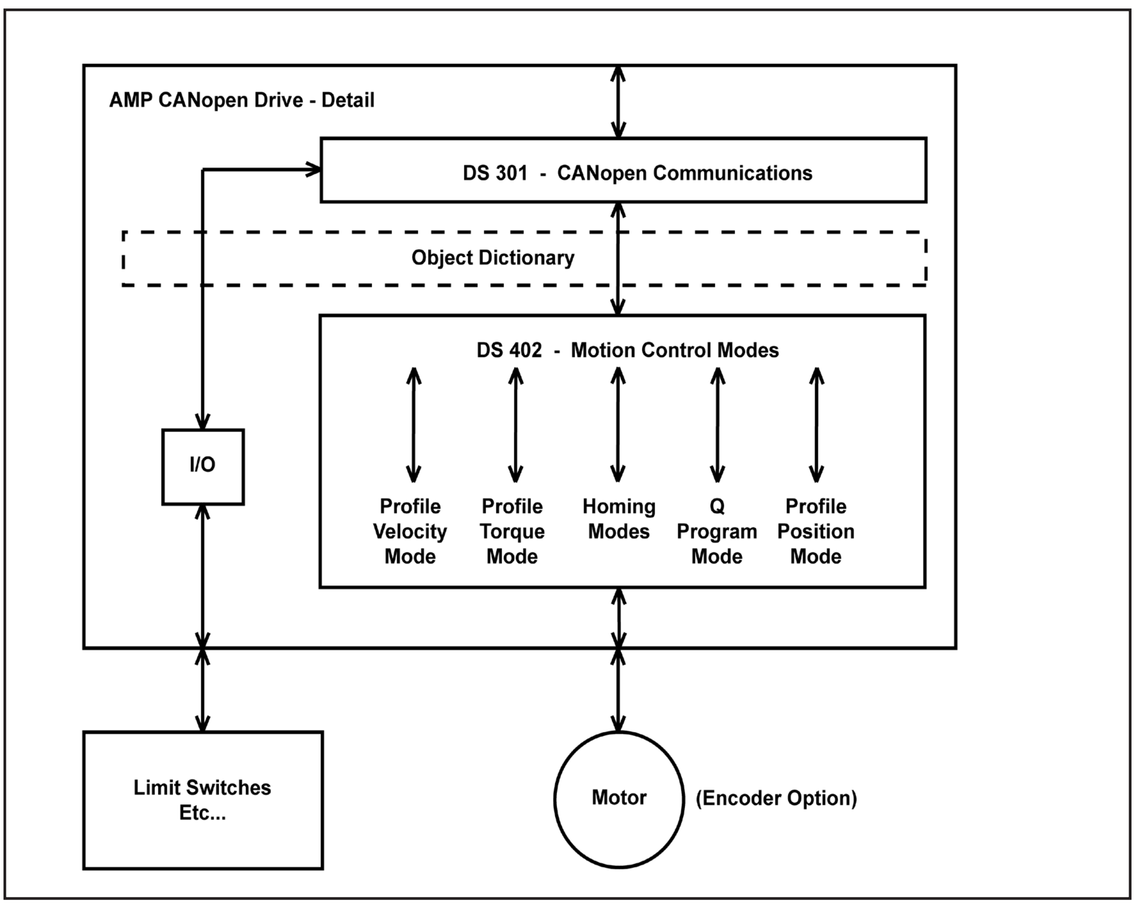


Figure 4: CANopen Drive - Motion Control Modes

Object Dictionary

The Object Dictionary is the core of any CANopen node. It provides links to all the communication and running parameters of a node. The Object Dictionary is defined in the EDS, which lists all supported objects, along with their sub-objects, if applicable.

Any Object Dictionary Entry may be accessed using the standard SDO protocol, while some may be accessed using the low-overhead PDO protocol.

Details: See Appendix I for a description of SDO and PDO Access.

Electronic Data Sheet

The EDS, available `on the website, lists all the properties of every supported object in the Object Dictionary.

Compatibility Issues

To maintain compatibility with the DSP402 spec, the following Object Dictionary entries are defined as 32-bit values, but not all the bits will be used by the drive. The entries below should be written and read as per the "CANopen Spec Length", but only the "Effective AMP Length" will be used.

For example, Object Dictionary Entry 0x606C (Velocity_Actual_Value) is defined as a 32-bit number, but only the lower 16 bits will be used by the drive. The upper 16 bits will be ignored, and should be left as zero when issuing a write command.

Details: See Appendix A for a description of Parameter Unit Scaling.

Object Dictionary Entry	Object Dictionary Index	CANopen Spec Length	Effective AMP Length
Velocity_Target_Value	0x606C	32	16
Profile_Velocity	0x6081	32	16
Profile_Acceleration	0x6083	32	16
Profile_Deceleration	0x6084	32	16
Homing Speeds (multiple)	0x6099 subs 1,2	32	16
Homing_Acceleration	0x609A	32	16
Drive Inputs	7003	32	8
Drive Outputs	60FE	32	8
Target_Velocity	0x60FF	32	16

Table 3: Object Dictionary Compatibility Issues

DS 301 OBJECT DESCRIPTIONS	COMMENT FIELDS
0x1000	Device Type
0x1001	Error Register
0x1005	COB - ID Sync
0x1008	Manufacturer Device Name
0x100A	Manufacturer Software Revision
0x100C	Guard Time
0x100D	Life Time Factor
0x1017	Producer Heart Beat
0x1018	Identity Object
0x1200	Server SDO Parameter 0
0x1400	Receive PDO Communications Parameter 0
0x1401	Receive PDO Communications Parameter 1
0x1402	Receive PDO Communications Parameter 2
0x1403	Receive PDO Communications Parameter 3
0x1600	Receive PDO Mapping Parameter 0
0x1601	Receive PDO Mapping Parameter 1
0x1602	Receive PDO Mapping Parameter 2
0x1603	Receive PDO Mapping Parameter 3
0x1800	Transmit PDO Communications Parameter 0
0x1801	Transmit PDO Communications Parameter 1
0x1802	Transmit PDO Communications Parameter 2
0x1803	Transmit PDO Communications Parameter 3
0x1A00	Transmit PDO Mapping Parameter 0
0x1A01	Transmit PDO Mapping Parameter 1
0x1A02	Transmit PDO Mapping Parameter 2
0x1A03	Transmit PDO Mapping Parameter 3

Table 4: DS Object Descriptions

Details: For a complete description see CANopen Specification DS301.

DSP 402 OBJECTS	COMMENT FIELD
0x603F	Error Code - See Appendix B: DSP Error Codes
0x6040	Control Word
0x6041	Status Word
0x605A	Quick Stop Option Code 2 & 6 Only
0x6060	See Section: Modes of Operation
0x6061	Modes of Operation Display
0x6064	Position_Target Value Calculated
0x606C	Velocity Target Value Calculated
0x6071	Target Torque Servos Only
0x6073	Running Current - See Appendix A: (For Scaling)
0x607A	Target Position
0x607C	Home Offset
0x6081	Profile Velocity - See Appendix A: (For Scaling)
0x6083	Profile Acceleration - See Appendix A: (For Scaling)
0x6084	Profile Deceleration - See Appendix A: (For Scaling)
0x6085	Quick Stop Deceleration - See Appendix A: (For Scaling)
0x6098	Homing Method
0x6099	Homing Speeds - See Appendix A: (For Scaling)
0x609A	Homing Acceleration - See Appendix A: (For Scaling)
0x60FE	Drive Outputs
0x60FF	Target Velocity - See Appendix A: (For Scaling)
0x6502	Supported Drive Modes

Table 5: DSP 402 Objects

MANUFACTURER SPECIFIC OBJECTS	COMMENT FIELD
0x7001	Home Switch Select 0x01 thru 0x06
0x7002	Idle Current - See Appendix A: (For Scaling)
0x7003	Display Drive Inputs
0x7007	Q Sequence Number Select 1 thru 12
0x7009	Velocity Actual Value - Calculated via Motor Encoder - Appendix A
0x700A	Position Actual Value - Calculated via Motor Encoder
0x700B	DSP Status Code - See Appendix B: (For Scaling)
0x700C	Acceleration_Current - See Appendix A: (For Scaling)
0x700D	Acceleration_Current - See Appendix A: (For Scaling)
0x700E	Analog Input 1
0x700F	Analog Input 2

Table 6: Manufacturer Specific Objects

General Purpose Registers

Applied Motion Products has provided twenty three general purpose registers, which may be accessed over CANopen. All twenty three registers are thirty two bit, Read/Write registers. They are volatile, so the information in these registers will not be saved after a power cycle.

The same twenty three registers may also be accessed and manipulated via a stored Q Program, if desired. Table 6, below, shows the cross-reference between the CANopen OD entry and the Q Programmer register address.

For more information on using the General Purpose registers in a Q Program, please see the Host Command Reference, available at http://www.applied-motion.com/sites/default/files/Host_Command_Reference.pdf.

Details: See Appendix G for a description of Q-Program Mode.

Register Name	OD Address	Q Register Address
User Defined Register 0	0x4000	0
User Defined Register 1	0x4001	1
User Defined Register 2	0x4002	2
User Defined Register 3	0x4003	3
User Defined Register 4	0x4004	4
User Defined Register 5	0x4005	5
User Defined Register 6	0x4006	6
User Defined Register 7	0x4007	7
User Defined Register 8	0x4008	8
User Defined Register 9	0x4009	9
User Defined Register 10	0x400A	:
User Defined Register 11	0x400B	;
User Defined Register 12	0x400C	<
User Defined Register 13	0x400D	=
User Defined Register 14	0x400E	>
User Defined Register 15	0x400F	?
User Defined Register 16	0x4010	@
User Defined Register 17	x04011	[
User Defined Register 18	0x4012	\
User Defined Register 19	0x4013	]
User Defined Register 20	0x4014	^
User Defined Register 21	0x4015	_
User Defined Register 22	0x4016	`

Table 7: User Defined Registers in CANopen and Q Program

Appendix A - Parameter Unit Scaling

The table below shows conversions from physical units in to the internal drive units. Use this table to scale parameters before they are passed to the drive. Units must be rounded to the nearest whole number, and represented in hexadecimal, before they are sent to the drive. Negative numbers should be expressed using the two's compliment representation.

Parameter Scaling Chart

Parameter Type	Multiplier	Units
Current	0.01	A
Velocity	0.0042	rps
Acceleration	0.1667	rps
Distance	1	step

Table 8: Parameter Scaling Chart

Examples

Querying the Point to Point Profile Acceleration from the drive:

SDO Read from OD 0x6083 returns a value of 0x226, or 550 decimal. Using the calculation below, we can see that this yields an acceleration of 91.85 rps.

550 * 0.1667 RPS = 91.685 RPS

Set the Point to Point Acceleration to 10rps:

10 RPS / 0.1667 = 59.988

Using the formula below, and rounding to the nearest whole number, we can see that we need to issue an SDO Write to OD 0x6083 with a value of 60 decimal, or 0x3C.

Set the Target Position to -2000 steps:

Because the relationship between physical steps and internal steps is one-to-one, we simply need to send the value -2000 to OD 0x607A. In order to send a negative number, we must use the "two's compliment" notation. To find the two's compliment, we must subtract the value 2000 from 2^{32}, since the Target Position is a 32 bit number.

2^{32} - 2000 = 4, 294, 965, 296 = 0xFFFFF830

If that's confusing, don't worry - an easier way would be to use the scientific mode of the Windows Calculator. Just enter -2000, and click on the "hex" button.

Appendix B - Response Codes

Table 9: Object 0X603F DSP Error Codes

Hex Value	SV	STAC6	ST	STM
0001	Limit Position	(Stall)		
0002	CCW Limit			
0004	CW Limit			
0008	***Over Temp***			
0010	***Internal Voltage***	***Excess Regen***	***Internal Voltage***	***Internal Voltage***
0020	***Over Voltage***			
0040	Under Voltage	***Under Voltage***	Under Voltage	Under Voltage
0080	***Over Current***			
0100	***Bad Hall Sensor***	***Open Motor Winding***		
0200	***Bad Encoder***			(not used)
0400	Comm Error			
0800	Bad Flash			
1000	Wizard Failed	No Move		
2000	Current Foldback	Motor Resistance Out of Range	(not used)	(not used)
4000	Blank Q Segment			
8000	No Move	(not used)		

NOTE: Items in ***bold italic*** represent Drive Faults, which automatically disable the motor. Use the OF command in a Q Program to branch on a Drive Fault.

Table 10: Object 0X700B DSP Status Codes

Hex Value	Status Code bit definition
0001	Motor Enabled (Motor Disabled if this bit = 0)
0002	Sampling (for Quick Tuner)
0004	Drive Fault (check Alarm Code)
0008	In Position (motor is in position)
0010	Moving (motor is moving)
0020	Jogging (currently in jog mode)
0040	Stopping (in the process of stopping from a stop command)
0080	Waiting (for an input)
0100	Saving (parameter data is being saved)
0200	Alarm present (check Alarm Code)
0400	Homing (executing an SH command)
0800	Wait Time (executing a WT command)
1000	Wizard running (Timing Wizard is running)
2000	Checking encoder (Timing Wizard is running)
4000	Q Program is running
8000	Initializing (happens at power up)

Appendix C - Profile Position Mode

General Mode Description

Profile Position Mode is a point-to-point operating mode. The mode operates on “set-points”, which consist of Velocity, Acceleration, Deceleration, and Target Position. Once all the set-point parameters have been set, the drive buffers all the commands and begins executing the set-point. In “set of set-points” mode, a new set-point can be sent to the drive while a set-point is executing.

Enable Profile Position Mode

To enable Profile Position Mode, the value 0x0001 must be written to the “Mode of Operation” Object Dictionary entry (OD), located at dictionary address 0x6060.

The mode of operation can be verified using OD 0x6061, “Mode of Operation Display”, which is updated when the current operation mode is accepted.

Set Running Parameters

Set the distance, velocity, acceleration, and deceleration using OD entries 0x607A, 0x6081, 0x6083, and 0x6084 respectively.

Starting/Stopping Motion

To indicate a new set point and start motion, toggle bit(4) by sending 0x001F to control word ODE 0X6040.

To enable drive operation, the value 0x001F must be written to the “Control Word” Object Dictionary entry (OD), located at dictionary address 0x6040. This puts the drive into “Operation Enabled” state, and signals that there is a new set point ready. The drive acknowledges the receipt of a valid set point using bit 12 of the Status Word (OD 0x6041).

Because the set point is edge-triggered, once the drive receives and processes the set point, the new set point of the control word must be cleared by writing 0x000F to the Control Word register.

While the drive is acting on a set point, a new set point may be entered and triggered using the new set point. The second set point will be received as soon as it is processed, or at the end of the previous set point, which ever is later.

Controlword Bits

New Setpoint (bit 4) - set this bit high to clock in a new set-point. Once the drive has accepted the set-point, it will respond by setting Statusword bit 12 high. Control Word bit 4 should then be taken low.

Change of Set-point (bit 9) - if this bit is low, the previous set-point will be completed and the motor will come to rest before a new set-point is processed. If bit 9 is high, the motor will continue at the speed commanded by the previous set-point until it has reached the position commanded by the previous set-point, then transition to the speed of the new set-point.

Change Set Immediately (bit 5) - if this bit is high, the new set-point will take effect immediately. The motor speed will transition to the speed commanded by the new set-point and will proceed to the position commanded by the new set-point.

Abs/rel (bit 6) - if this bit is high, the set-point distance is relative. For example, if the previous motor position was 10,000 steps and a new set-point is issued with a distance of 20,000, the final position will be 30,000. If bit 6 is low, the distance is absolute. If the previous motor position was 10,000 and a new set-point is issued with a distance of 20,000, the new position will be 20,000. (The distance travelled from the previous position to the new position will be 10,000 steps.) For best results, do not change this bit while the motor is moving.

Details: See DSP402-2 Profile Position Mode.

Note: Two set points can be set up, but if status bit 12 is high, then the buffer is full and another set point will be ignored.

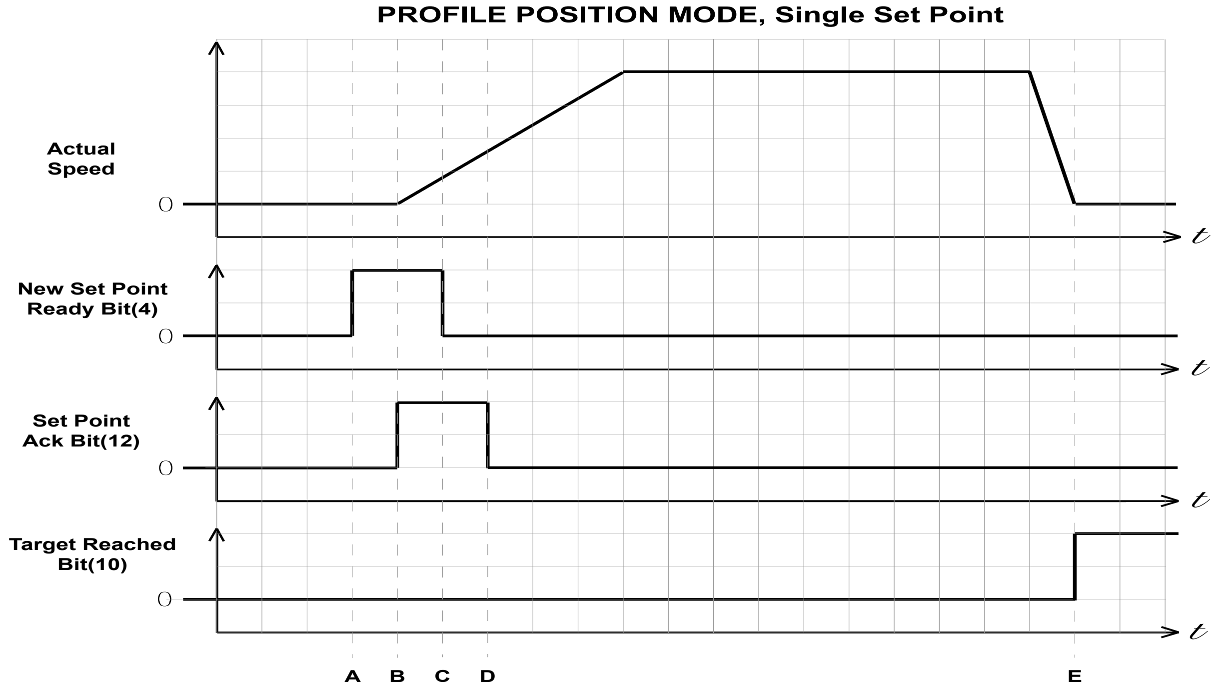


Figure 5: Single Set Point

Graph Point	New Set Point Ready Bit	Set Point Acknowledge Bit	Target Reached Bit	What's Going On
Start	0	0	0	Drive waiting for set-point
A	0 -> 1	0	0	User tells drive a set-point is ready
B	1	0 -> 1	0	Drive acknowledges set-point, and starts executing the set-point
C	1 -> 0	1	0	User pulls New Set Point Ready bit low
D	0	1 -> 0	0	Drive pulls set point acknowledge bit low, indicating that it is ready to receive another set point

Table 11: Single Set-Point Profile Position Move

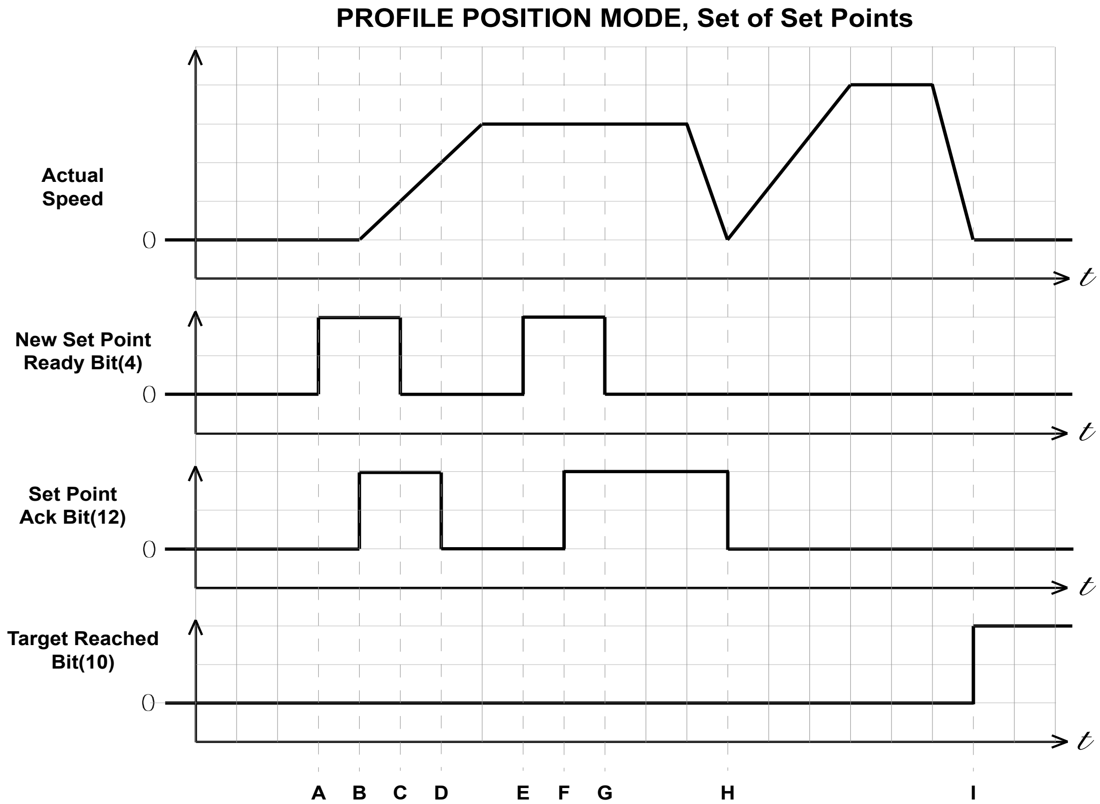


Figure 6: Multiple Set Points, Stopping Between Moves

In this example, controlword bits 9 (Change of Set-point) and 5 (Change Set Immediately) are 0. The motor comes to rest between moves.

Graph Point	New Set Point Ready Bit	Set Point Ack Bit	Target Reached Bit	What's Going On
Start	0	0	0	Drive waiting for set-point
A	0 -> 1	0	0	User tells drive a set-point is ready
B	1	0 -> 1	0	Drive acknowledges set-point, and starts executing the set-point
C	1 -> 0	1	0	User pulls New Set Point Ready bit low
D	0	1 -> 0	0	Drive pulls set point ack bit low, indicating that it is ready to receive another set point
E	0 -> 1	0	0	User tells drive another set-point is ready
F	1	0 -> 1	0	Drive acknowledges set-point, and buffers it, since another set-point is still in progress
G	1 -> 0	1	0	User pulls New Set Point Ready bit low
H	0	1 -> 0	0	Drive pulls set point ack bit low, and starts executing the new set-point as soon as the old one is finished

Table 12: Multi-Set-Point Profile Position Move

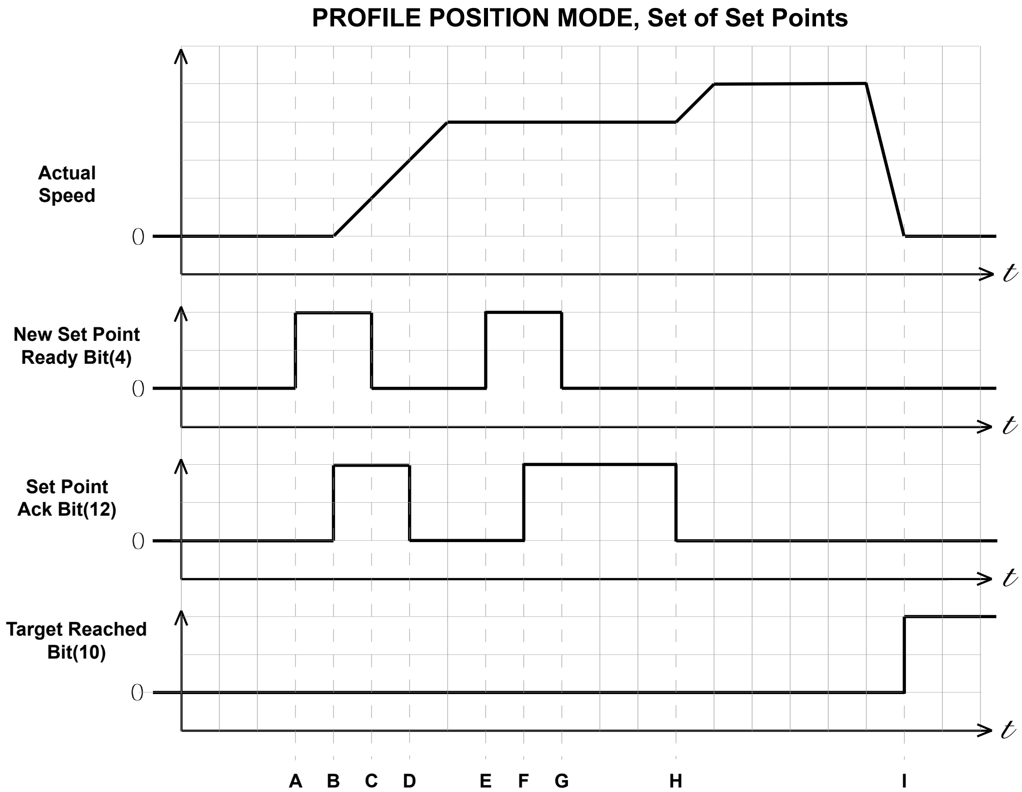


Figure 7: Multiple Set Points, Continuous Motion

In this example, controlword bit 9 (Change of Set-point) is 1 and control word bit 5 (Change Set Immediately) is 0. The motor continues at the speed of the first set-point until is reaches the distance of the first set-point, then changes to the new set-point speed. The motion is continuous.

Graph Point	New Set Point Ready Bit	Set Point Ack Bit	Target Reached Bit	What's Going On
Start	0	0	0	Drive waiting for set-point
A	0 -> 1	0	0	User tells drive a set-point is ready
B	1	0 -> 1	0	Drive acknowledges set-point, and starts executing the set-point
C	1 -> 0	1	0	User pulls New Set Point Ready bit low
D	0	1 -> 0	0	Drive pulls set point ack bit low, indicating that it is ready to receive another set point
E	0 -> 1	0	0	User tells drive another set-point is ready
F	1	0 -> 1	0	Drive acknowledges set-point, and buffers it, since another set-point is still in progress
G	1 -> 0	1	0	User pulls New Set Point Ready bit low
H	0	1 -> 0	0	Drive pulls set point ack bit low, and starts executing the new set-point as soon as the old one is finished
I	0	0	1	The set-point is finished, and there are no set-points in the buffer, so the Target Reached bit is set

Table 13: Multi-Set-Point Profile Position Move with Continuous Motion

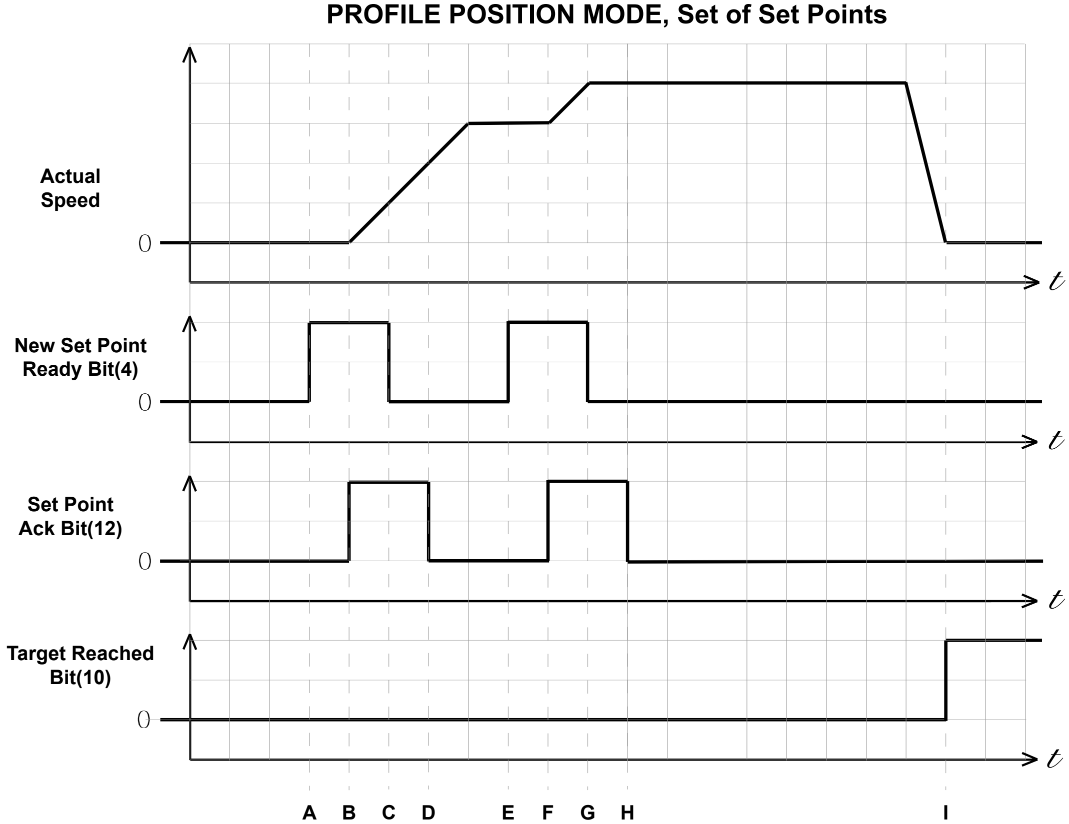


Figure 8: Multiple Set Points, Immediate Change in Motion

In this example, controlword bit 9 (Change of Set-point) is 1 and control word bit 5 (Change Set Immediately) is 1. The motor immediately changes to the new set-point speed without completing the first set-point. The motion is continuous.

Graph Point	New Set Point Ready Bit	Set Point Ack Bit	Target Reached Bit	What's Going On
Start	0	0	0	Drive waiting for set-point
A	0 -> 1	0	0	User tells drive a set-point is ready
B	1	0 -> 1	0	Drive acknowledges set-point, and starts executing the set-point
C	1 -> 0	1	0	User pulls New Set Point Ready bit low
D	0	1 -> 0	0	Drive pulls set point ack bit low, indicating that it is ready to receive another set point
E	0 -> 1	0	0	User tells drive another set-point is ready
F	1	0 -> 1	0	Drive acknowledges set-point, and immediately executes it, beginning the transition to the new set-point speed and position
G	1 -> 0	1	0	User pulls New Set Point Ready bit low
H	0	1 -> 0	0	Drive pulls set point ack bit low.
I	0	0	1	The set-point is finished, and there are no set-points in the buffer, so the Target Reached bit is set

Table 14: Multi-Set-Point Profile Position Move with Immediate Change in Motion

Appendix D - Profile Velocity Mode

General Mode Description

Profile Velocity mode is a relatively simple operating mode. Once the velocity, acceleration, and deceleration are set, the drive will either command the motor to accelerate to the running velocity according to the acceleration parameter, or halt movement according to the deceleration parameter.

Figure 7, below, shows an example of Profile Velocity Mode. The top graph shows the actual speed of the motor. The middle graph shows the target speed value, and the bottom graph shows the halt bit in the Control Word.

Table 12, below, explains how the Halt Bit and Target Velocity may be used together to affect motor speed. Notice that between Points B and C, the motor does not come to a complete stop, rather, it decelerates according to the Profile Deceleration value starting at Point B, and when the halt bit transitions at Point C, it accelerates immediately back to the target speed. Also notice at Point E, reducing the Target Speed to zero has the same effect as enabling the Halt Bit, since the drive is commanding the motor to move at zero speed.

It should be noted that both enabling the Halt Bit, and setting the Target Velocity to zero keep torque applied to the motor. In order to allow the shaft to move freely, the NMT state must be put in to the “Drive Disabled” state.

Enable Profile Velocity Mode

To enable Profile Velocity Mode, the value 0x0003 must be written to the “Mode of Operation” Object Dictionary entry (OD), located at dictionary address 0x6060.

The mode of operation can be verified using OD 0x6061, “Mode of Operation Display”, which is updated when the current operation mode is accepted.

Set Running Parameters

Set the velocity, acceleration, and deceleration using OD entries 0x60FF, 0x6083, and 0x6084 respectively.

Enable Drive Operation

To enable drive operation, the value 0X010F must be written to the “Control Word” Object Dictionary entry (OD), located at dictionary address 0x6040. This puts the drive into “Operation Enabled” state, with the motion halted.

Starting/Stopping Motion

To start and stop motion, toggle the Control Word register’s HALT bit (bit 8). When HALT = 0, motion starts or continues; when HALT = 1, motion stops. The bit can be toggled by writing 0X010F and 0x000F to the Control Word, (OD0x6040).

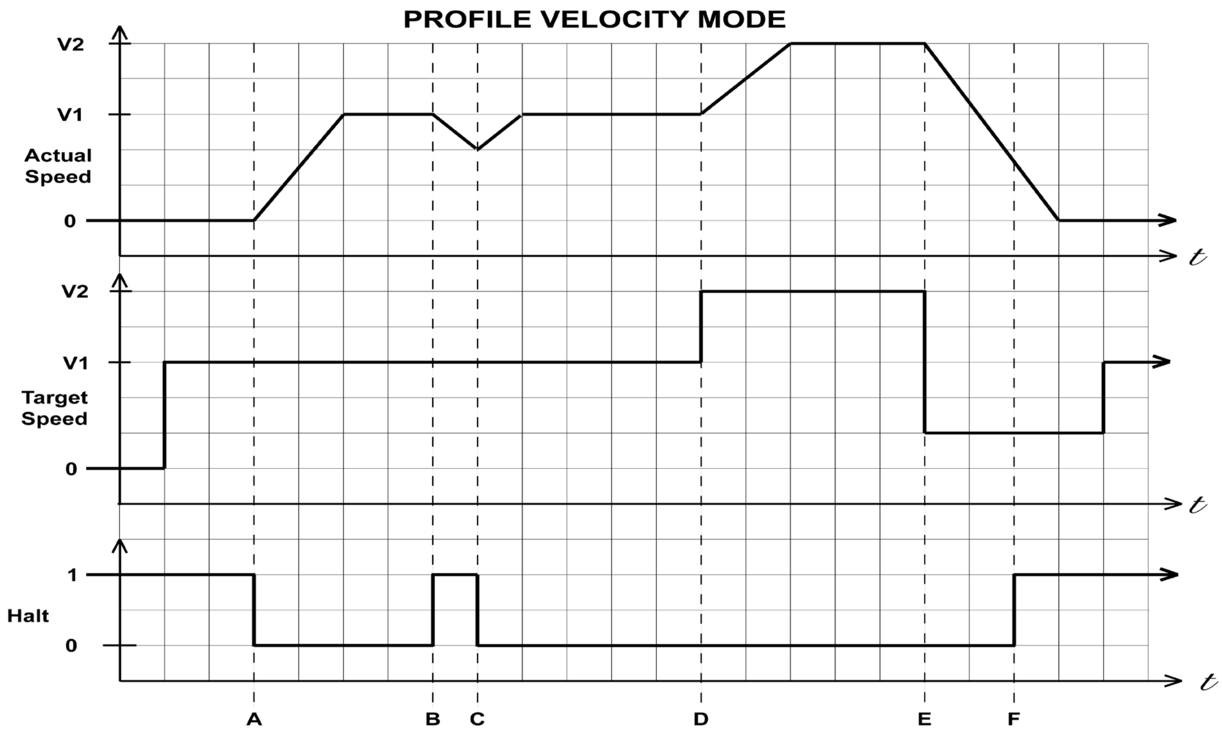


Figure 9: Profile Velocity Mode

Graph Point	Target Speed	Halt Bit	Drive command to Motor
Start	0	1	Motor stopped
A	V1	1 -> 0	Motor accelerate to speed V1
B	V1	0 -> 1	Motor decelerate to stopped
C	V1	1 -> 0	Motor accelerate to V1
D	V1 -> V2	0	Motor accelerate from V1 to V2
E	V2 -> 0	0	Motor decelerate from V2 to 0
F	0	0 -> 1	Motor remains stopped
G	0 -> V1	1	Motor remains stopped

Table 15: Profile Velocity Mode Example

Appendix E - Homing Mode

Set Running Parameters

Set the homing and index velocities, acceleration/deceleration, offset and home sensor (if required) using OD entries 0x6099, 0x609A, 0x607C, and 0x7001 respectively.

Note: It is important that the limit switch settings have been defined in ST Configurator or QuickTuner prior to using the CANopen Homing Mode.

Enable Homing Mode

To enable Homing Mode, the value 0x0006 must be written to the "Mode of Operation" Object Dictionary entry (OD), located at dictionary address 0x6060. The mode of operation can be verified using OD 0x6061, "Mode of Operation Display", which is updated when the current operation mode is accepted.

To put the drive into Operation Enabled Mode, write 0x000F to the Control Word (OD 0x6040).

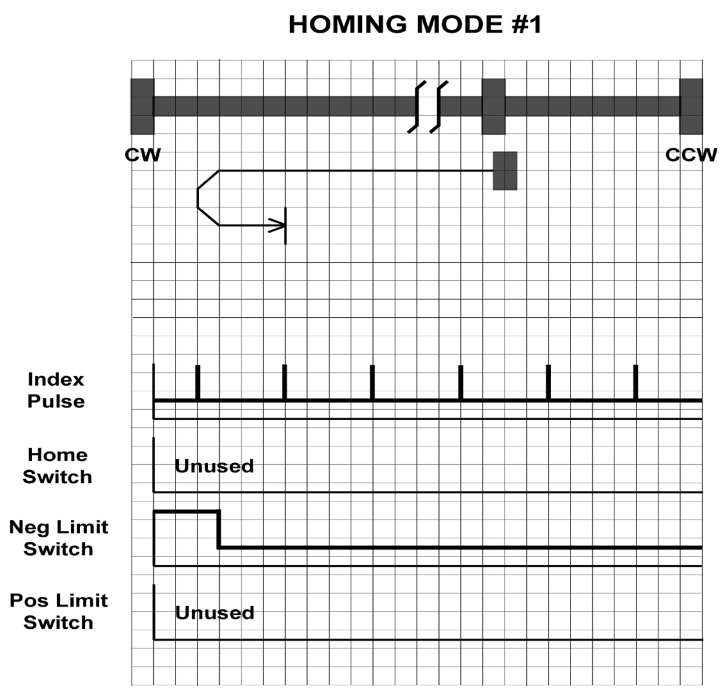
Starting the Homing Procedure

Set the Homing Method required using OD entry 0x6098. To start the homing procedure, the bit four of the Control Word Object Dictionary entry located at dictionary address 0x6040, must transition from 0 to 1. The status of the homing procedure can be monitored using the Status Word (OD 0x6041).

Homing Mode Diagrams

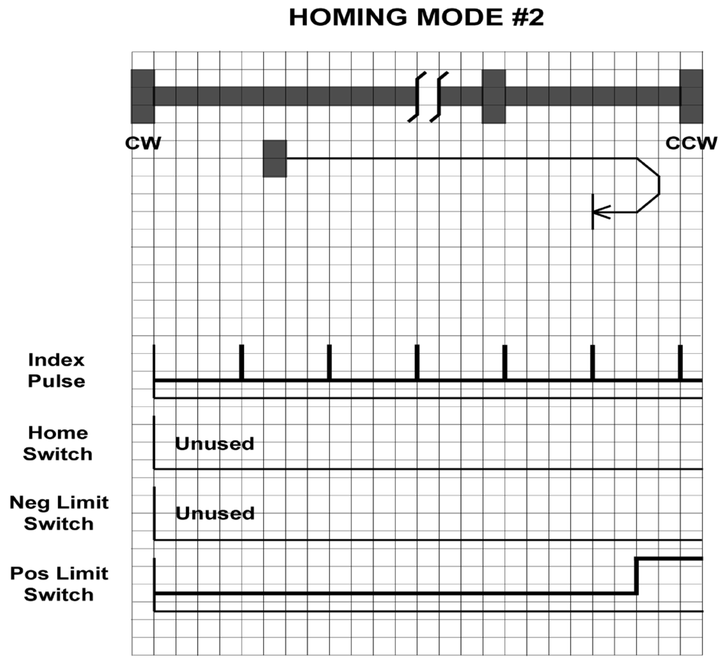
Homing Mode 1

Homing method 1, as shown below, homes to the first index CCW after the CW limit switch is reached:



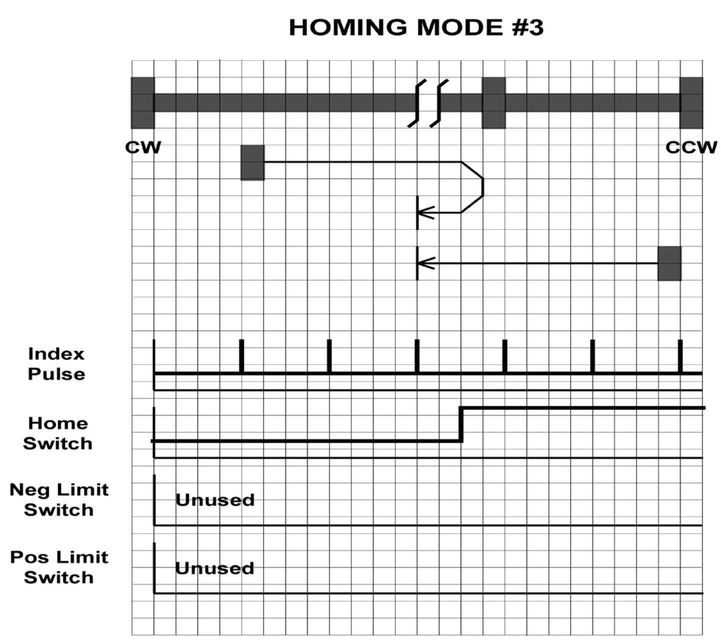
Homing Mode 2

Homing method 2 homes to the first index CW after the CCW limit switch is reached:



Homing Mode 3

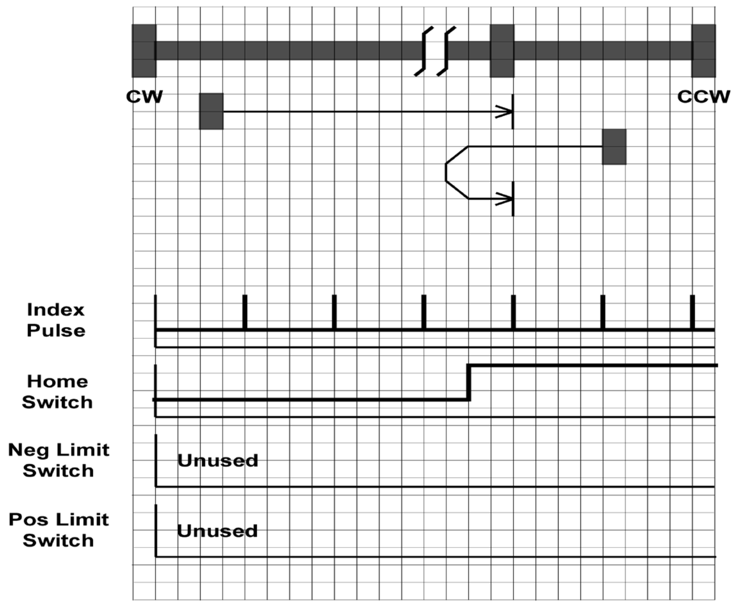
Homing method 3 homes to the first index CW after the positive home switch changes state. The initial direction of motion is dependent on the state of the home switch:



Homing Mode 4

Homing method 4 homes to the first index CCW after the positive home switch changes state. The initial direction of motion is dependent on the state of the home switch:

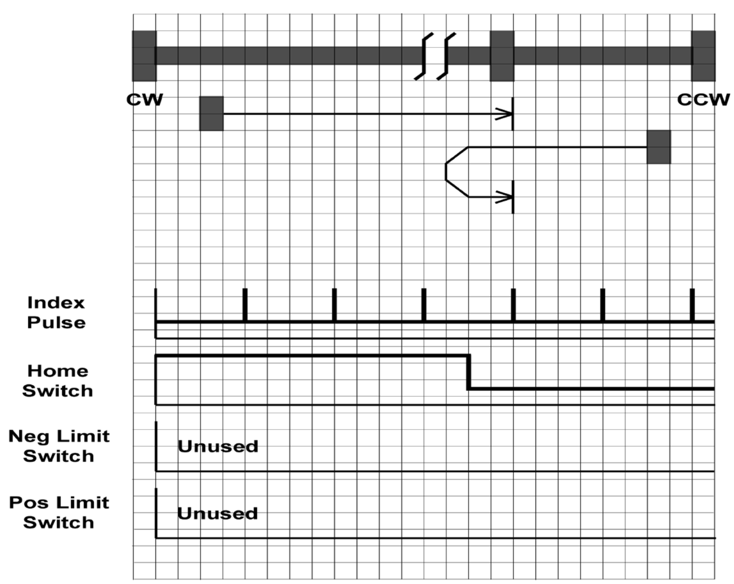
HOMING MODE #4



Homing Mode 5

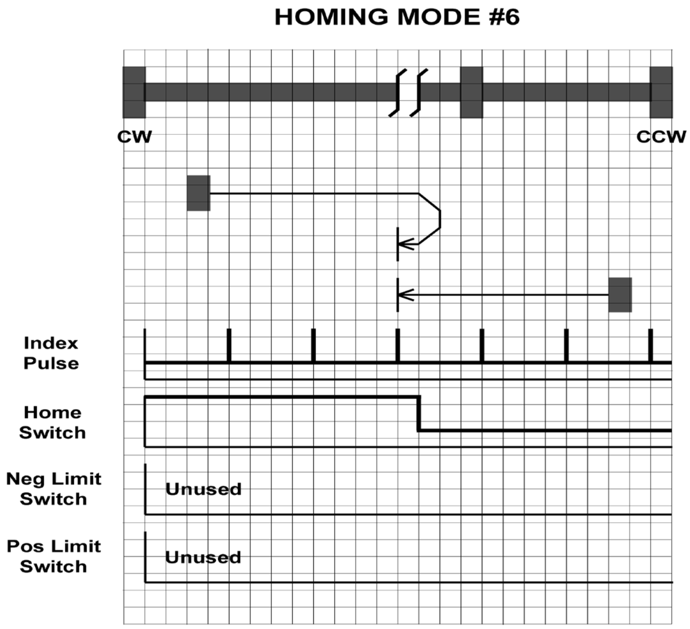
Homing method 5 homes to the first index CCW after the negative home switch changes state. The initial direction of motion is dependent on the state of the home switch:

HOMING MODE #5



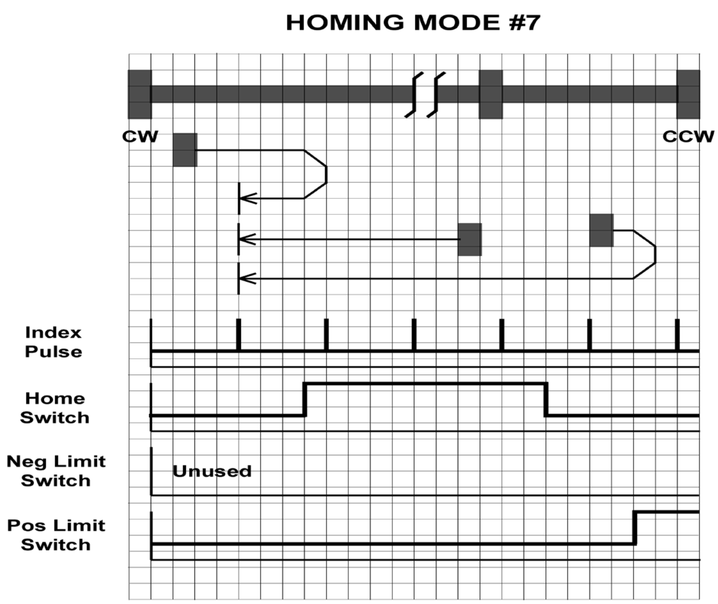
Homing Mode 6

Homing method 6 homes to the first index CW after the negative home switch changes state. The initial direction of motion is dependent on the state of the home switch:



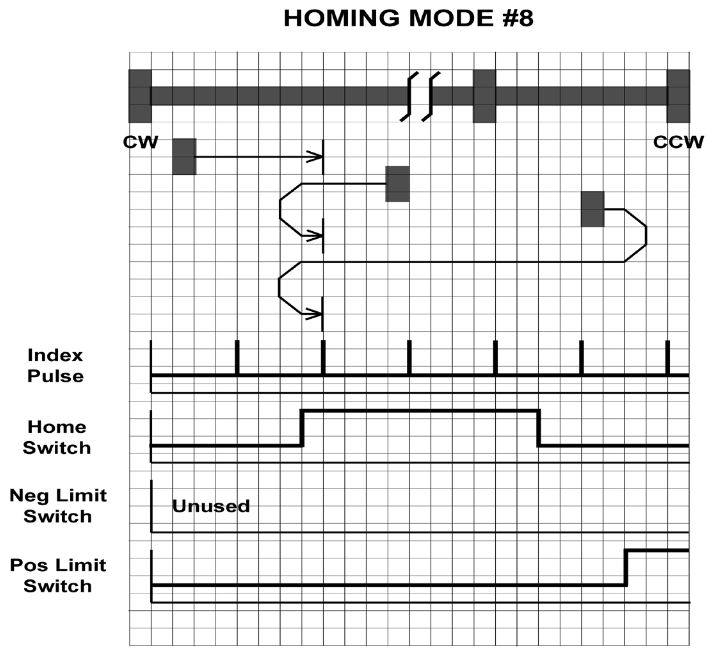
Homing Mode 7

Homing method 7 starts moving CCW (or CW if the home switch is active), and homes to the first index CW of the home switch transition:



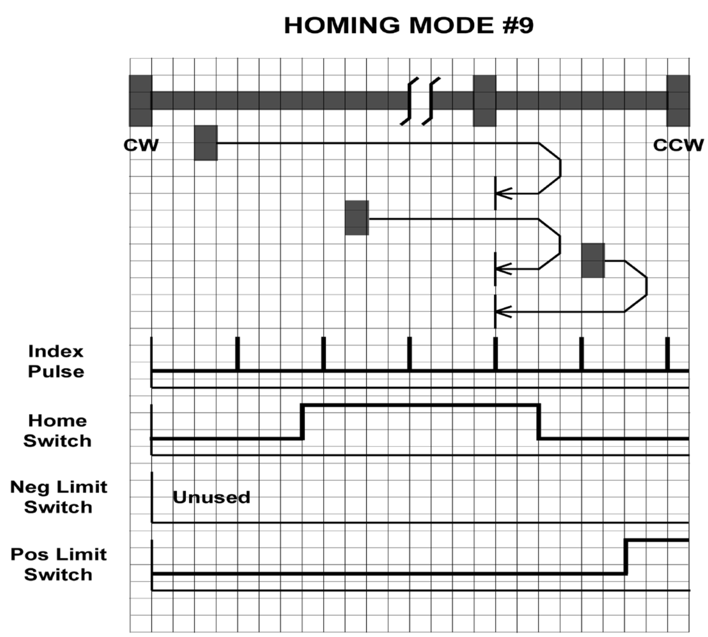
Homing Mode 8

Homing method 8 starts moving CCW (or CW if the home switch is active), and homes to the first index CCW of the home switch transition:



Homing Mode 9

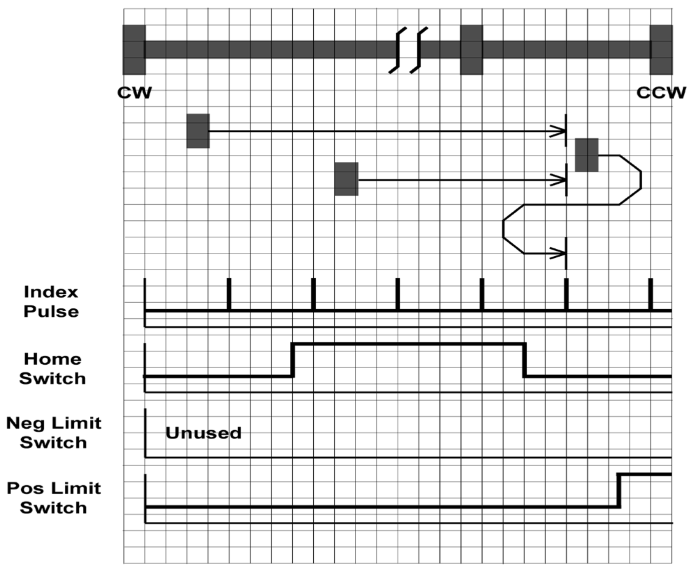
Homing method 9 starts moving CCW and homes to the first index CW of the home switch transition:



Homing Mode 10

Homing method 10 starts moving CCW and homes to the first index CCW of the home switch transition:

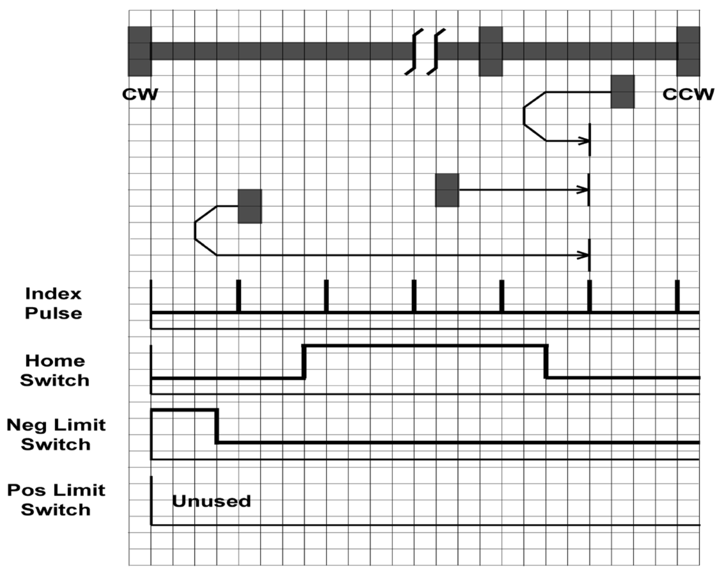
HOMING MODE #10



Homing Mode 11

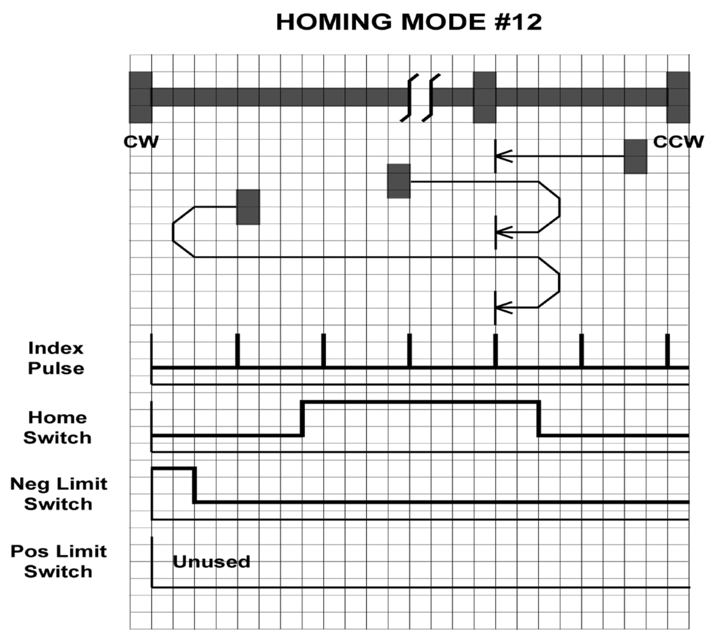
Homing Method 11 starts moving CW (or CCW if the home switch is active), and homes to the first index CCW of the home switch transition:

HOMING MODE #11



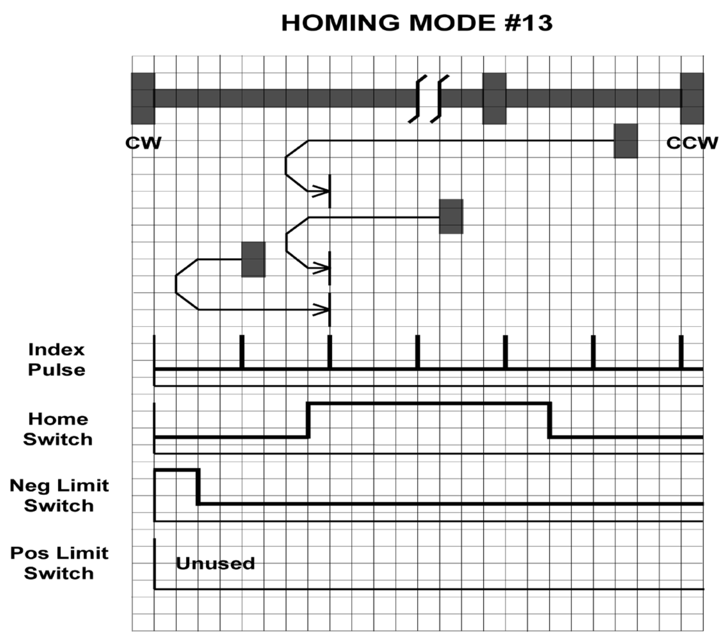
Homing Mode 12

Homing Method 12 starts moving CW (or CCW if the home switch is active), and homes to the first index CW of the home switch transition:



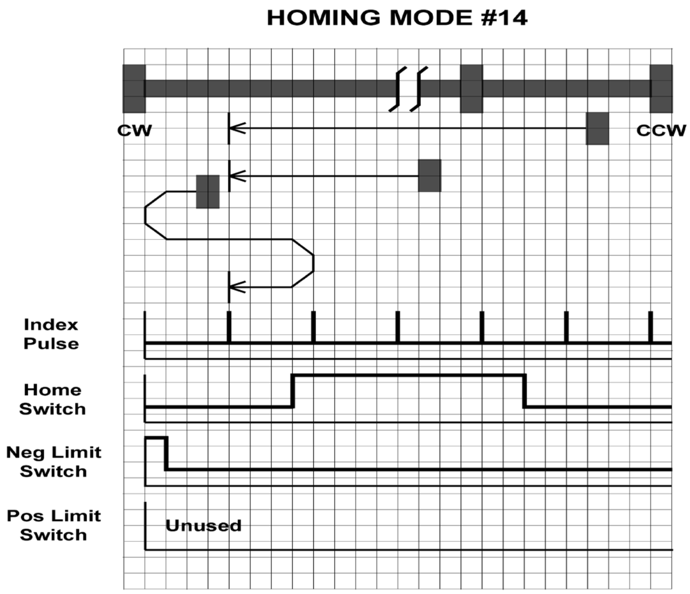
Homing Mode 13

Homing method 13 starts moving CW and homes to the first index CCW of the home switch transition:



Homing Mode 14

Homing method 14 starts moving CW and homes to the first index CW of the home switch transition shown above.

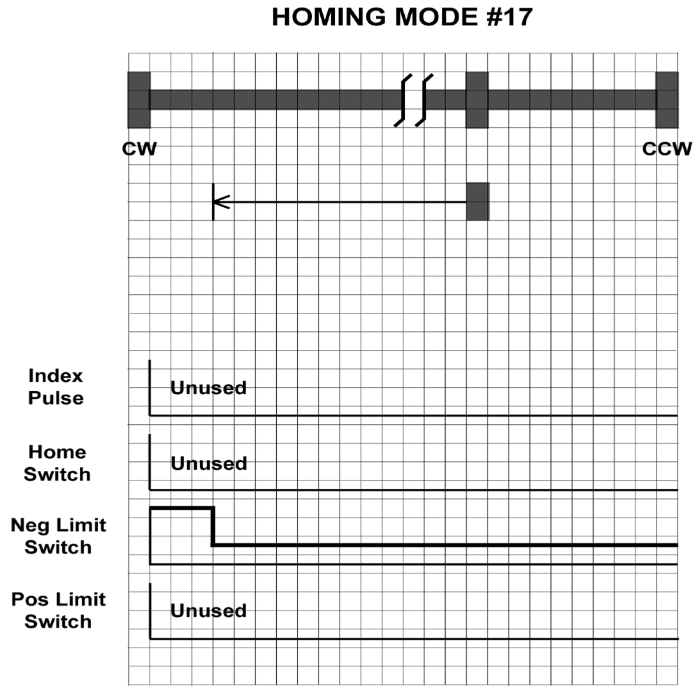


Homing Modes 15 and 16

Homing Modes 15 and 16 are reserved for future expansion.

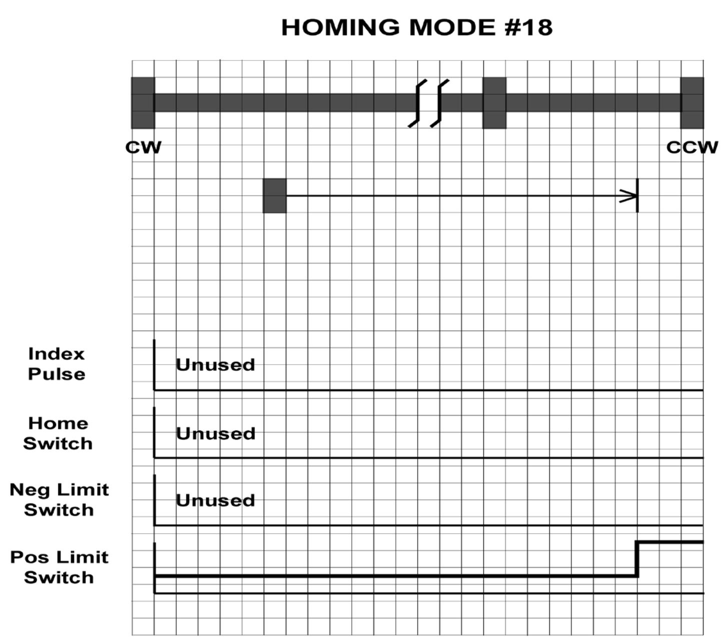
Homing Mode 17

Homing method 17 homes to the CW limit switch



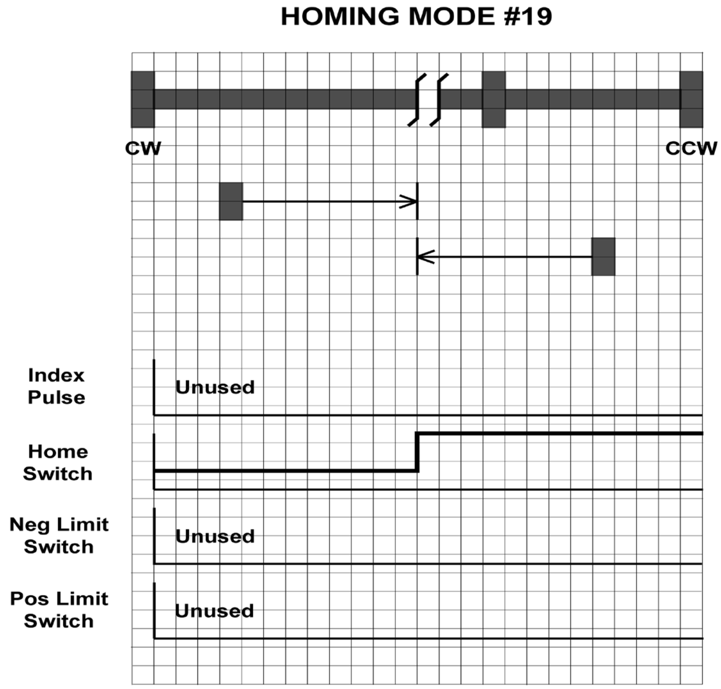
Homing Mode 18

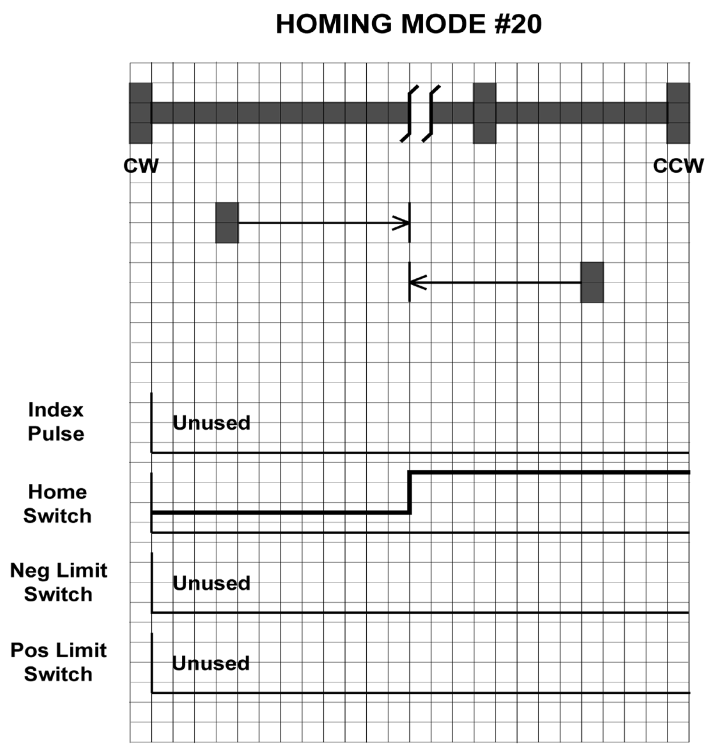
Homing method 18 homes to the CCW limit switch:



Homing Modes 19 and 20

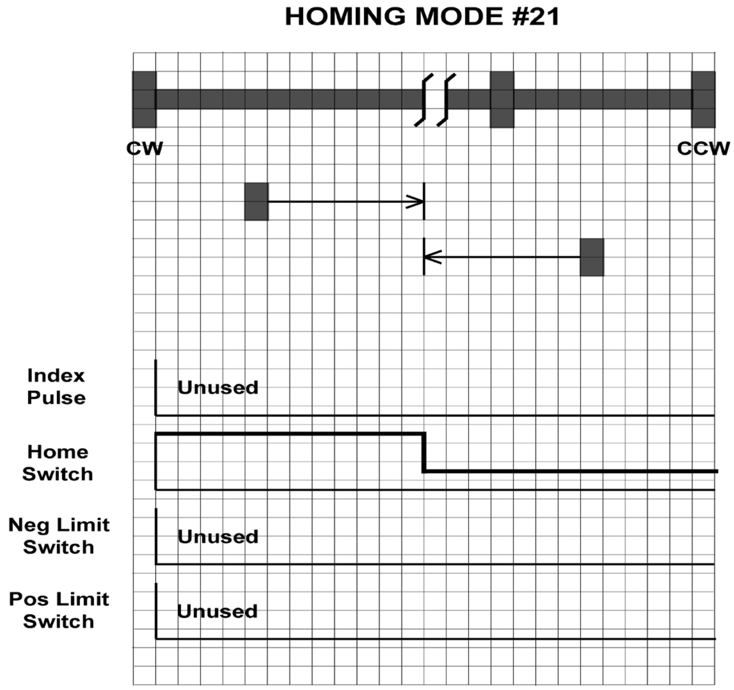
Homing methods 19 and 20 home to the home switch transition:

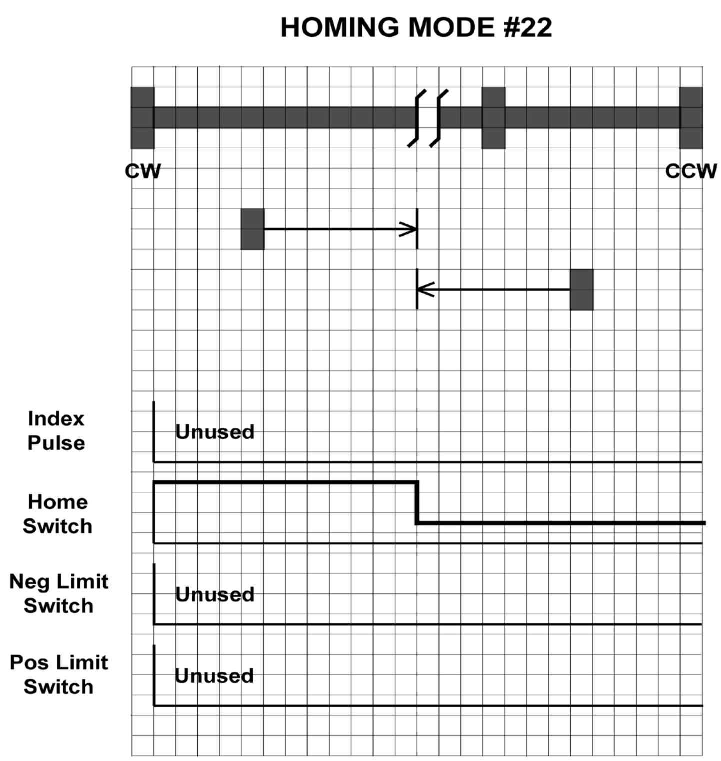




Homing Modes 21 and 22

Homing methods 21 and 22 home to the home switch transition:

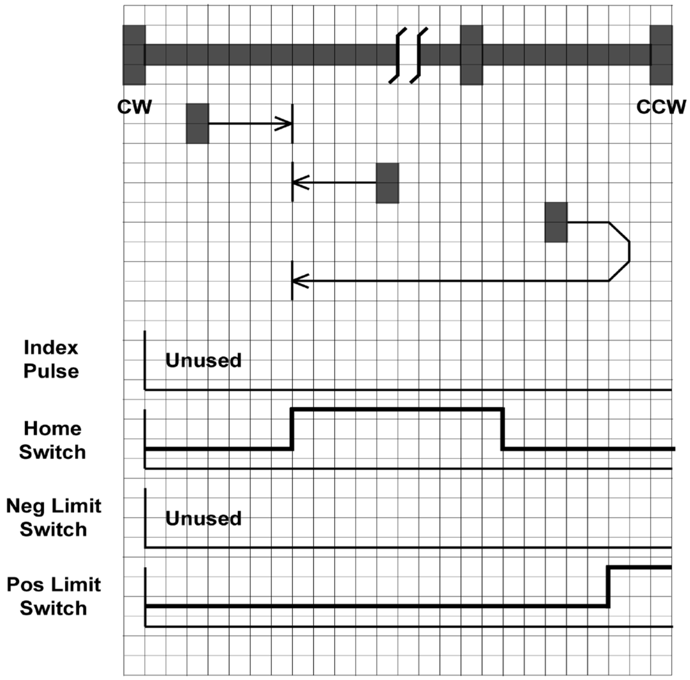




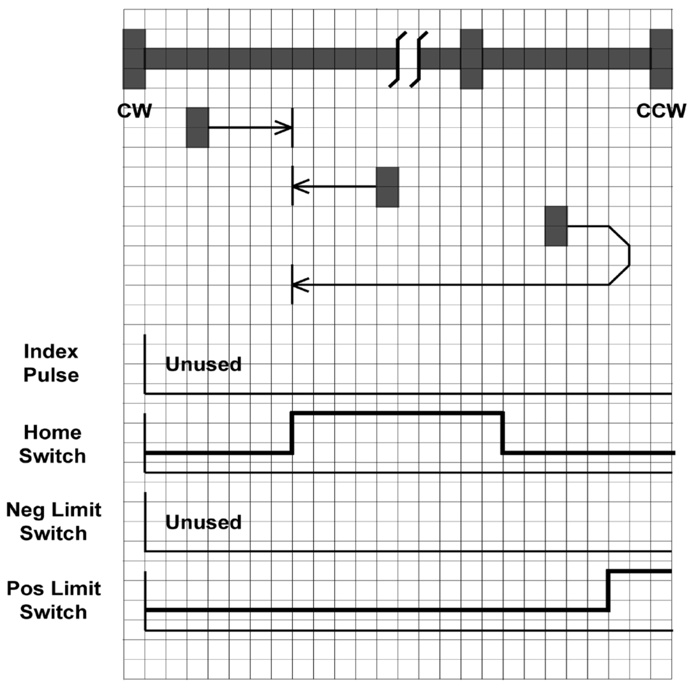
Homing Modes 23 and 24

Homing methods 23 and 24 home to the home switch transition shown below, and "bounce off" the CCW limit, if required.

HOMING MODE #23

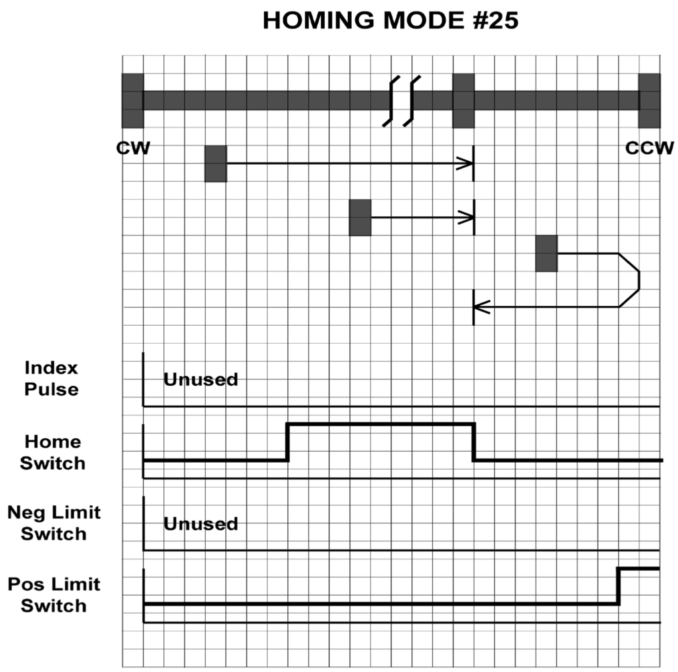


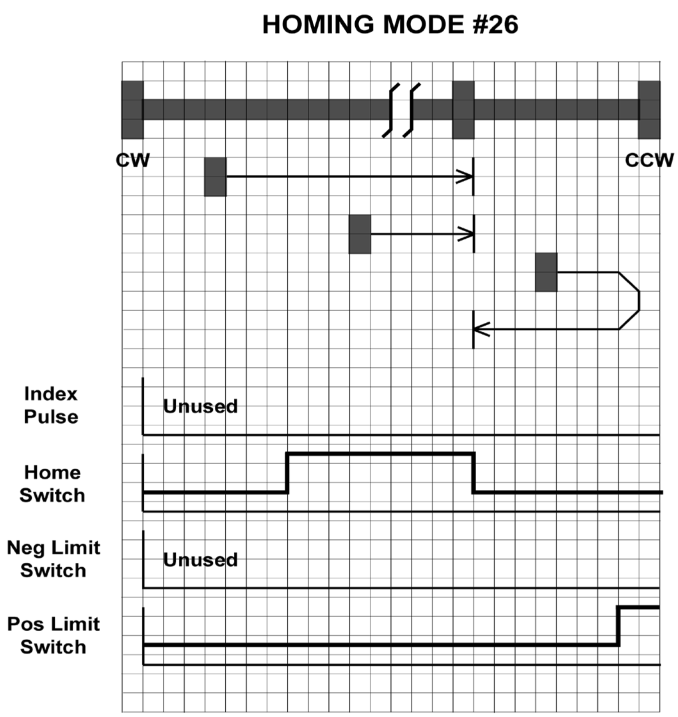
HOMING MODE #24



Homing Modes 25 and 26

Homing methods 25 and 26 home to the home switch transition shown below, and “bounce off” the CCW limit, if required.

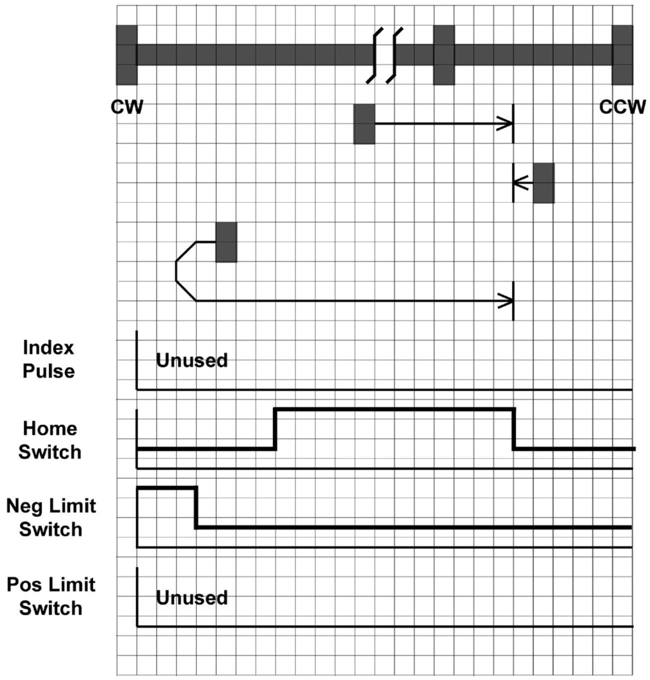




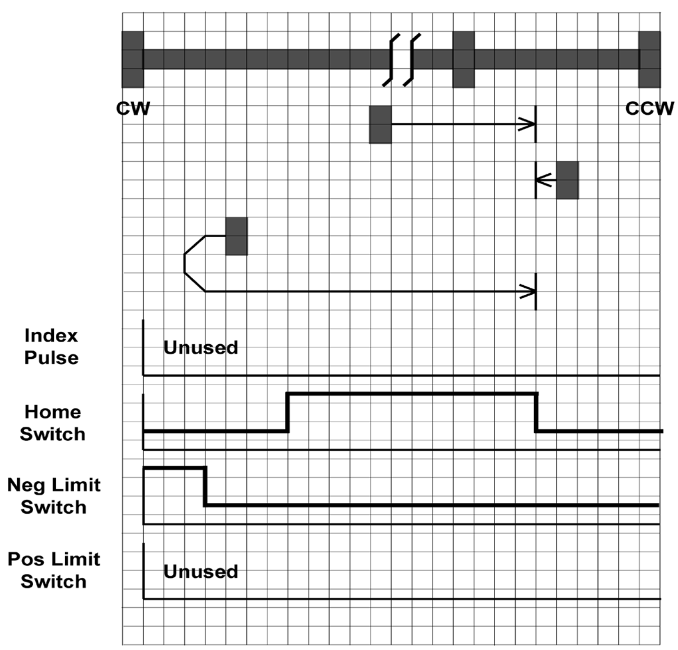
Homing Modes 27 and 28

Homing methods 27 and 28 home to the home switch transition shown below, and “bounce off” the CW limit, if required.

HOMING MODE #27



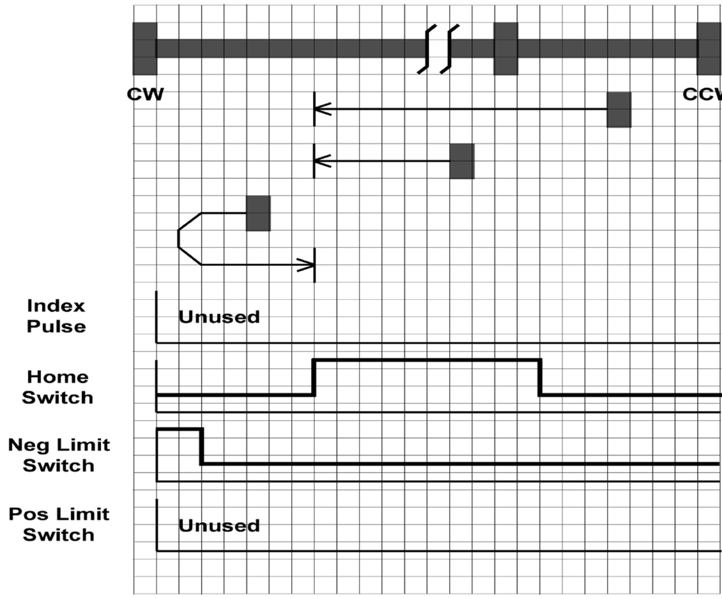
HOMING MODE #28



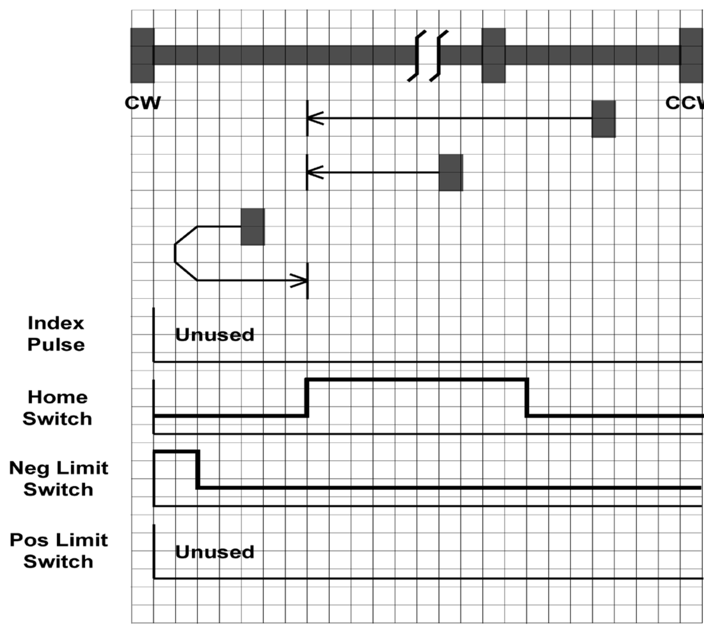
Homing Modes 29 and 30

Homing methods 29 and 30 home to the home switch transition shown below, and "bounce off" the CW limit, if required.

HOMING MODE #29



HOMING MODE #30



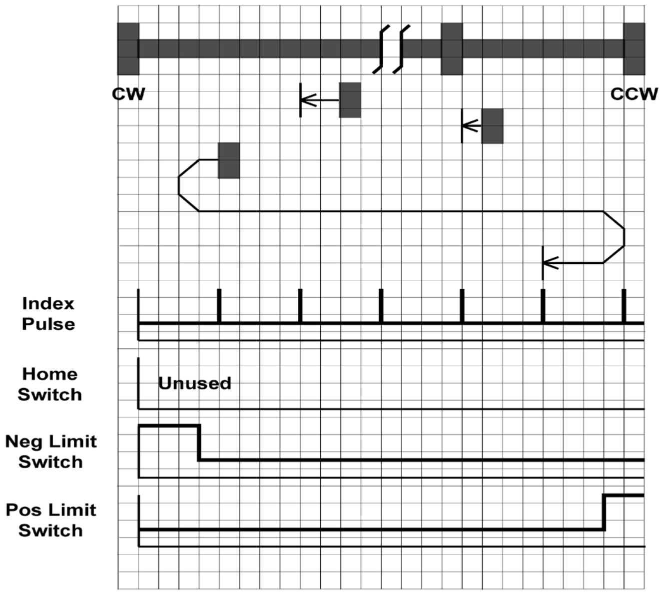
Homing Modes 31 and 32

Homing Modes 31 and 32 are reserved for future expansion.

Homing Mode 33

Homing method 33 homes to the next index pulse CW from the current position. If the CW limit is hit, the drive resets to the CCW limit, and continues searching for a limit in the CW direction:

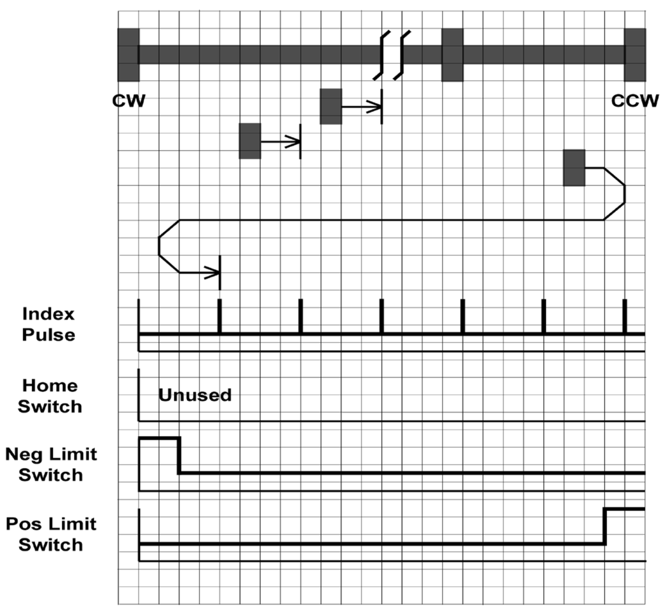
HOMING MODE #33



Homing Mode 34

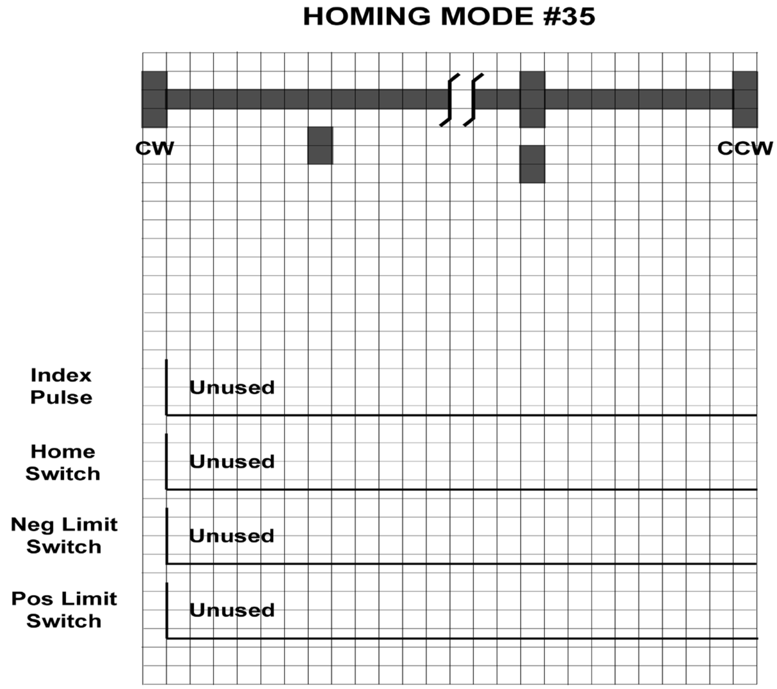
Homing method 34 homes to the next index pulse CCW from the current position. If the CCW limit is hit, the drive resets to the CW limit, and continues searching for a limit in the CCW direction:

HOMING MODE #34



Homing Mode 35

Homing method 35 takes the current position to be the home position. In this mode, the Home Offset value is ignored, and the motor does not move at all:



Appendix F - Profile Torque Mode (Servo Only)

General Mode Description

Profile Torque mode is a servo-control torque operating mode. It requires knowledge of the Torque Constant of the motor, in Nm/A. If using an Applied Motion Products motor, this information may be found on our website, www.applied-motion.com. If using another motor, please refer to the motor print.

Enable Profile Torque Mode

To enable Profile Torque Mode, the value 0x0004 must be written to the "Mode of Operation" Object Dictionary entry (OD), located at dictionary address 0x6060.

The mode of operation can be verified using OD 0x6061, "Mode of Operation Display", which is updated when the current operation mode is accepted.

Set Running Parameters

The following parameters must be set:

Parameter Name	Object Dictionary Entry	Length (in bytes)	Units	Description
Torque Constant	0x7005	2	$\frac{m \cdot Nm}{a}$	Motor parameter, found on the motor print
Target Torque	0x6071	2	$m \cdot Nm$	Torque to be applied to the motor
Torque Slope	0x6087	4	$\frac{m \cdot Nm}{sec}$	Rate at which to ramp torque to new target

Figure 10: Set Running Parameters

Enable Drive Operation

To enable drive operation, the value 0x000F must be written to the "Control Word" Object Dictionary entry, located at dictionary address 0x6040. This puts the drive into "Operation Enabled" state, with no torque applied.

It should be noted that both enabling the Halt Bit, and setting the Target Torque to zero both ramp down the torque applied to the motor according to the Torque Slope. At the end of the slope, no torque will be applied to the motor, allowing the shaft to move freely.

Starting/Stopping Torque

To start and stop motion, toggle the Control Word register's HALT bit (bit 8). When HALT = 0, motion starts or continues; when HALT = 1, motion stops. The bit can be toggled by writing 0x010F and 0x000F to the Control Word register.

Parameter Calculations

Assume we want to apply a torque of 50 oz-in to an Applied Motion Products V0200-214-B-000 Valueline motor, with an SV7 drive, and a torque slope of 25oz-in/sec. From the Applied Motion Products website, we see that the Torque Constant of the motor is 0.07Nm/A. We must first convert the Nm/A constant given into mNm/A, as required by the Torque Constant Object Dictionary entry.

$$0.07\ \frac{Nm}{A} * 1000\ \frac{mA}{A} = 70\ \frac{m \cdot Nm}{A}$$

Because the drive works primarily in Nm, we must first convert 50 oz-in into Nm, using the conversion factor 141.6 oz-in/Nm.

$$\frac{50\ oz \cdot in}{141.6\ \frac{oz \cdot in}{Nm}} = 0.3531\ Nm$$

Now, we must take the required torque of 0.3531Nm and convert it into mNm, as required by the Target Torque Object Dictionary entry.

$$0.3531\ Nm * 1000\ \frac{m \cdot Nm}{Nm} = 353.1\ m \cdot Nm$$

This gives us a value of 353 mNm, rounded to the nearest whole number, for the Target Torque Object Dictionary Entry.

Finally, we must convert the slope from the given units of oz-in/sec into the required units of mNm/sec.

$$\left(25\ \frac{oz \cdot in}{sec}\right) * \left(\frac{1\ Nm}{141.6\ oz \cdot in}\right) * \left(\frac{1000\ m \cdot Nm}{1\ Nm}\right) = 176.55\ \frac{m \cdot Nm}{sec}$$

Rounding to the nearest whole number gives us a Torque Slope of 177 mNm/sec.

Current Verification

It is important to check the current that will be required of the drive, to ensure that it is within the limits of the servo amplifier. The SV7 drive, for example, has a continuous rating of seven amps, and a peak current of fourteen amps, which may be held continuously for two seconds. This means that a current of seven amps can be held indefinitely, and currents between seven and fourteen amps may be used in short bursts.

We can check the current draw from the example above using the target torque and the torque constant, as shown.

$$\frac{0.3531\ Nm}{0.07\ \frac{Nm}{A}} = 5.0443\ A$$

The resultant current, 5.0443A, is below the 7A continuous current rating of the SV7 drive, and well below the peak current rating of 14A.

With regard to the peak current, it is possible for the drive to maintain a current of 7A indefinitely, and peak up to 14A for up to two seconds continuously. Values between 7A and 14A may be held proportionally long.

More Information

More information on Profile Torque Mode can be found in the CANopen spec, DSP402-2.

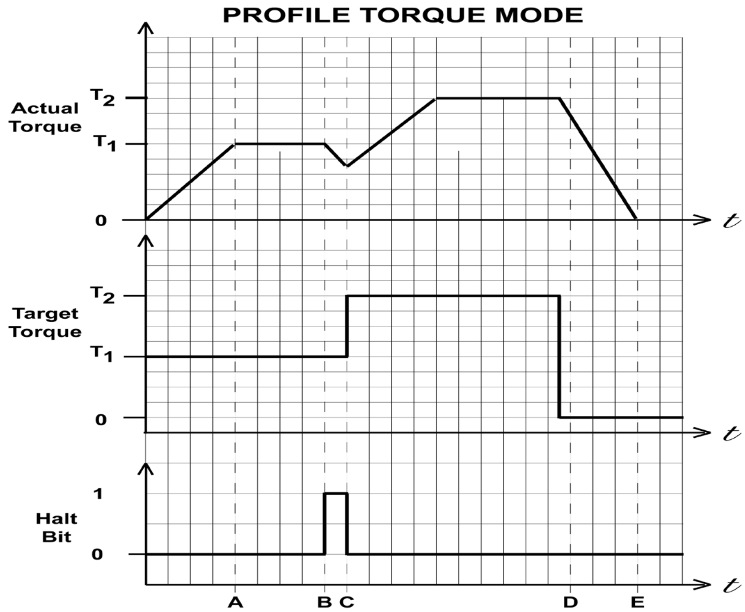


Figure 11: Profile Torque Mode

Graph Point	Target Torque	Halt Bit	Drive command to Motor
Start	T1	0	Ramp torque to T1
A	T1	0	Maintain torque at T1
B	T1	0 -> 1	Ramp Torque to zero
C	T1 -> T2	1 -> 0	Ramp Torque to T2
D	T2 -> 0	0	Ramp torque to zero
E	0	0	Maintain torque at zero

Table 16: Profile Torque Mode Example

Appendix G - Q Program Mode

General Mode Description

In order to expand the functionality of Applied Motion Products' CANopen drives, the Q programming language may be used to execute complex motion profiles that may not be possible within the scope of DSP402. The Q program must be pre-loaded in to the CANopen drive using Q Programmer (v1.3.5 or later).

Q Programs may also access and manipulate the CANopen General Purpose registers for use in stored programs. The section "General Purpose Registers" has a conversion chart between the OD entry and Q address.

Loading a Q Program

As with ST Configurator and QuickTuner, the drive must be powered up with the RS232 port connected, and the program running, for the CANopen drive to delay the normal boot-up procedure. The CAN boot-up can be resumed by closing the Q Programmer application, or by power-cycling the drive with the RS232 port disconnected.

Once Q Programmer has control of the drive, it may be used in the same way as any other Applied Motion Products Q drive. Please see the Q Command Reference for more information on Q programming.

Normal Q Program Execution

To execute a stored Q program on a single drive, the value of -1 (0xFF) must be written to the Mode of Operation register, located at dictionary address 0x6060.

The mode of operation can be verified using OD 0x6061, Mode of Operation Display, which is updated when the current operation mode is accepted.

Next, the desired Q segment number, 1-12, must be written to the Q Segment Number register, located at address 0x7007.

To enable drive operation, the value 0x000F must be written to the Control Word Object Dictionary entry (OD), located at dictionary address 0x6040. This puts the drive into Operation Enabled state, ready to run the Q program.

To run the selected Q program, the value of 0x001F must be written to the Control Word. The Q program will then run to completion. The Q program may be re-executed by a 0->1 transition of the Q Program Start Bit (bit 4) in the Control Word. To halt execution of a Q program, set the Halt Bit (bit 8) of the Control Word to 1. The Q program will halt immediately, and start from the beginning the next time a 0->1 transition is seen on the Q Program Start Bit after the Halt Bit has been cleared.

Synchronous Q Program Execution

To execute a stored Q program on a single drive, the value of -2 (0xFE) must be written to the Mode of Operation register, located at dictionary address 0x6060.

The mode of operation can be verified using OD 0x6061, Mode of Operation Display, which is updated when the current operation mode is accepted.

Next, the desired Q segment number, 1-12, must be written to the Q Segment Number register, located at address 0x7007. To enable operation 0x001F must be written to the control word ODE"6040".

Because we want to run the drive based on the SYNC pulse, we must set the SYNC pulse in the drive. We must set the COB-ID SYNC register, located at 0x1005. A standard value for the SYNC pulse is 0x80, but any unused COB-ID may be used. Refer to DS301 for a list of reserved COB-IDs.

Once the SYNC pulse has been set, and the desired Q segment has been set, the drive will execute the Q segment every time a SYNC pulse is sent. In this way, multiple drives may be instructed to start a Q program in a single, network-wide instruction.

To halt execution of a Q program, set the Halt Bit (bit 8) of the Control Word to 1. The Q program will halt immediately, and start from the beginning the next time a SYNC pulse is sent after the Halt Bit has been cleared.

Example Program

See the test program included on the CD that came with your Applied Motion Products CANopen drive.

More Information

More information on COB-IDs can be found in the CANopen spec, DS301.

More information on Q programming may be found in the Host Command Reference found in a PDF download at http://www.applied-motion.com/sites/default/files/Host_Command_Reference.pdf

More information on the General Purpose Registers may be found in the “General Purpose Registers” section of this manual.

Appendix H - Understanding NMT States

Under normal operation conditions, the NMT state machine will power up into the Initialization state, send out a boot-up packet, move into the Pre-Operational state, and start sending out heartbeats with the Pre-Operational state status code.

NMT Mode	NMT Control Command	NMT Status Code (Heartbeat)
Initialization/Node Reset	129	0
Pre-Operational	128	127
Operational	1	5
Stopped	2	4

Table 17: Understanding NMT States

Example: Building a CANopen NMT Data Frame

Assume we want to send a broadcast message to all CANopen nodes, to set them in to the Operational NMT state. The COB-ID for NMT commands is always 0. This ensures that an NMT command has the highest priority on the bus, and may never get preempted, except by another node sending out an NMT command.

The first data byte of an NMT command contains the NMT Control Command, which is 1 (Operational) in this case.

The second data byte contains either the Node ID of a target Node, or, in the event that the NMT master is requesting that all nodes change their NMT Mode, a zero. Because we are sending a broadcast message, we will use zero. The completed data frame is below.

COB ID	Data Length	Data Byte 0	Data Byte 1
0	2	1	0

Table 18: Example - NMT Data Frame

Appendix I - SDO and PDO Access

Enable SDO Use

To enable SDO Use, the NMT state must be either "Pre-Operational" or "Operational". Send a NMT message to put the node into either state. When completed, the heartbeat should return either 127 (Pre-Operational) or 5 (Operational). The drive is now ready to read and/or write all OD entries.

Example: Building an SDO Read Data Frame

Assume we want to read the Heartbeat time of Node 0x2E. We must send SDO Read request to the drive. The default COB ID for SDO requests is 0x600 (from DS301), plus the Node ID 0f 0x2E; this results in a specific COB ID for this message of 0x62E. The first data byte is reserved for the control byte, which is always 0x40 for an SDO Read. The next two bytes are reserved for the OD entry address, in Little Endian format. Because we want OD entry 1017, we stuff data byte 1 with 0x18 and data byte 2 with 0x10. Data byte three is reserved for the sub-index of the OD entry, which in this case is zero. The last four bytes are unused for SDO reads. Now we have the whole message, which looks like this:

COB ID	Data Length	Data Byte 0	1	2	3	4	5	6	7
0x62E	8	0x40	17	10	0x00	0x00	0x00	0x00	0x00
						DATA BYTES			

Table 19: Enable SDO Use

The drive will respond with a message with COB ID 0x580 + Node ID, or 0x5AE.

Examples

See the test program included on the CD that came with your Applied Motion Products CANopen drive.

More Information

More information on the SDO protocol can be found in the CANopen spec, DS301.

PDO Access

Enable PDO Use

To enable PDO Use, the NMT state must be set to "Operational". Send a NMT message to enable the Operational state. When completed, the heartbeat should return a 5. The drive is now ready to receive RPDOs, and will transmit TPDOs depending on the Transmission Type.

TPDO Transmission Types

There are several triggering options for Transmit PDOs, which are controlled by OD entries 0x1800 – 0x1803, and associated sub-entries.

Possible TPDO Triggers
SYNC pulse; Node will send TPDO after receiving one or multiple SYNC pulses.

Event/Timer; Node will issue TPD based on an internal event or timer.
Remote Request; Node will send TPDO after a remote request.

0	PDO transmitted on the next SYNC pulse after the Statusword has changed.
1	PDO transmitted on every SYNC pulse
2-240	PDO transmitted on every n SYNC pulses
254-255	PDO transmitted every time Statusword changes, or the Event Timer has expired

Table 20: TPDO Transmission Types

PDO Mapping - Stepper

PDO Name	First Mapped Parameter	OD Entry	# Bytes	Second Mapped Parameter	OD Entry	# Bytes	# Bytes Total
TPDO1	Status Word	6041	2				2
TPDO2	Status Word	6041	2	Target Position	6064	4	6
TPDO3	Status Word	6041	2	Target Velocity	606C	2	4
TPDO4	Input Status	7003	1				1
RPDO1	Control Word	6040	2				2
RPDO2	Control Word	6040	2	Target Distance	607A	4	6
RPDO3	Control Word	6040	2	Target Velocity	60FF	2	4
RPDO4	Output State	60FE	1				1

Table 21: PDO Mapping - Stepper

PDO Mapping - Servo

PDO Name	First Mapped Parameter	OD Entry	# Bytes	Second Mapped Parameter	OD Entry	# Bytes	# Bytes Total
TPDO1	Status Word	6041	2				2
TPDO2	Status Word	6041	2	Actual Position	0x700A	4	6
TPDO3	Status Word	6041	2	Actual Velocity	0x7009	2	4
TPDO4	Input Status	7003	1				1
RPDO1	Control Word	6040	2				2
RPDO2	Control Word	6040	2	Target Distance	607A	4	6
RPDO3	Control Word	6040	2	Target Velocity	60FF	2	4
RPDO4	Output State	60FE	1				1

Table 22: PDO Mapping - Servo

PDO COB ID

Because PDOs are directly mapped to Object Dictionary entries, no overhead is required when working with them. RPDOs may be sent directly, with the COB ID being the default RPDO COB ID plus the node ID. For example, the default RPDO1 COB ID is 0x200. Therefore, the COB ID for RPDO1 to Node 0x2E would be 0x200 + 0x02E = 0x22E. The default COB IDs for each PDO may be found in DS301, on page 78.

Example: Building an RPDO Data Frame

Assuming we want to set the Controlword of Node 0x2E to 0x7E4F, we see that we can use RPDO1 to accomplish this. We already know the COB ID will be 0x22E, and from the mapping table above, we know that the first two message bytes will contain the Controlword. Remembering Endianness, the first data byte will be 0x4F and the second date byte will be 0x7E. Now we have the whole message, which looks like this:

COB ID	Data Length	Data Byte 0	Data Byte 1
0x22E	2	0x4F	0x7E

Table 23: PDO COB IDs

Examples

See the test program included on the CD that came with your Applied Motion Products CANopen drive.

More Information

More information on PDO mapping can be found in the CANopen spec, DSP402-3.

More information on the PDO protocol can be found in the CANopen spec, DP301.

Applied Motion Products

404 Westridge Drive

Watsonville, CA 95076

USA

tel / 831-761-6555

fax / 831-761-6544

www.applied-motion.com