GLENTEK INC.

SERVO DRIVES AND SERVO MOTORS

CANopen® Communication

User Manual

2nd Edition 2023

www.glentek.com

Preface

Glentek was founded in Glendora, California in 1964. One year later, we incorporated and moved to Santa Monica, California. In the beginning, our company designed and manufactured servo drives and related electronics for the defense industry, including applications involving advanced tactical aircraft and ship-based radar systems. During the 1970's we began production on some of the first PWM servo drives for the growing CNC machinery market. Further growth resulted in another move in 1980, this time to our current location in El Segundo, California.

We expanded into the medical market, producing custom servo drives for both CAT Scan and PET scan machines. To provide the highest performance and reliability possible, we began designing and manufacturing a complete line of servo motors to accompany our servo drives in the mid 1980's, providing a complete package for OEMs. Our first foray in DSP-based drives and the development of MotionMaestro, our Windows-based set-up and tuning software, occurred in the 1990's. Since then, we have released new drives capable of communicating using the latest protocols and standards, first Synqnet, then CANopen, and now EtherCAT. Our latest drives also accept feedback from the latest devices, including analog sin/cos encoders and absolute serial encoders.

For more information and details on our products, visit the company's website at www.glentek.com.

Contents

Tables and Figures

1. Scope

CANopen is an open standard embedded machine control protocol. CAN is a serial communication interface. The CANopen protocol is developed for the CAN physical layer. In this document, CAN is reserved for physical layer description, while CANopen refers to the communication protocol. An end-user should be able to read through the manual and be able to communicate with and operate Glentek's CANopen drives and motors.

Glentek's CANopen drives follow CiA 301 communications protocol; IEC 61800-7-301 communication's profile, and IEC 61800-7-201 device profile for drives and motion control. The International Electrotechnical Commission (IEC) is an international standards organization that publishes international standards for electronics. CAN in Automation (CiA) is a non-profit organization that develops and supports CAN-based higher-layer protocols. Additional information regarding the physical CAN layer and protocol can be found in the CiA 301, IEC 61800-7-301, and IEC 61800-7-201(CiA 402). They can be found at the website www.iec.ch and www.can-cia.org.

For further technical questions, please visit our website at www.glentek.com or call us at (310) 322-3026.

2. Terms and Abbreviations

ACK- Acknowledgement
C- Constant
CiA- CAN in Automation
COB-ID- Communication Object Identifier
CRC- Cyclic Redundancy Check
CSP- Cyclic Synchronous Position
CSV- Cyclic Synchronous Velocity
FIFO- First-in-First-out
IDE- Identifier Extension
IP- Interpolated Position
NLS- Negative Limit Switch
NMT- Network management
Node- A device, drive, motor, switch, connected to a CANopen interface
PDO- Process data objects
PLS- Positive Limit Switch
PV- Profile Velocity
RO- Read Only
RPDO- Receive PDO
RTR- Remote Transmission Request
RW- Read and Write
RX- Receive
SDO- Service data objects
TX- Transmit
TPDO- Transmit PDO

3. CANopen Objects

3.1 Overview

CANopen is a CAN-based protocol developed for flexible embedded network applications. Ultimately, CANopen provides a protocol that standardizes communication between devices and applications across different manufacturers and industries. CANopen devices must have an object dictionary. The object dictionary defines various communication protocols and device profiles to specify the functionality of the drive. Objects in the object dictionary are grouped according to the structure outlined in Table 3-1: Object Dictionary Structure. Three frequently used object types are the Communication Profile Area, Glentek-Specific Profile Area, and the Standardized Device Profile. The Communication Profile Area contains communication specific parameter common to all CANopen devices. The Glentek-Specific Profile Area contains parameters specific to Glentek drives. The Standardized Device Profile Area contains all data objects common to a class of CANopen devices that may be read or written via the network.

Table 3-1: Object Dictionary Structure

Index Range	Object Type
0001h – 001Fh	Static Data Types
0020h – 003Fh	Complex Data Types
0040h – 005Fh	Glentek-Specific Complex Data Types
0060h – 025Fh	Device Profile Specific Data Types
1000h – 1FFFh	Communication Profile Area
2000h – 5FFFh	Glentek-Specific Profile Area
6000h – 67FFh	Standardized Device Profile

Each object on the object dictionary is defined by a 16-bit address called the Object Index. Some objects may have an 8-bit Sub-Index to address a subcomponent in the object. The Object Index and the Sub-Index are used in CANopen messages to transfer data and ultimately operate slave nodes on a CAN Bus. Figure 3-1: CAN Network depicts a CAN Master along with 3 slave nodes connected to the CAN BUS. Each slave node has a unique Node-ID. The CAN Master uses CANopen Messages addressed to the unique Node-IDs as well as information about Object Index and Sub-Index to control and get status data of all the slave nodes on the bus.

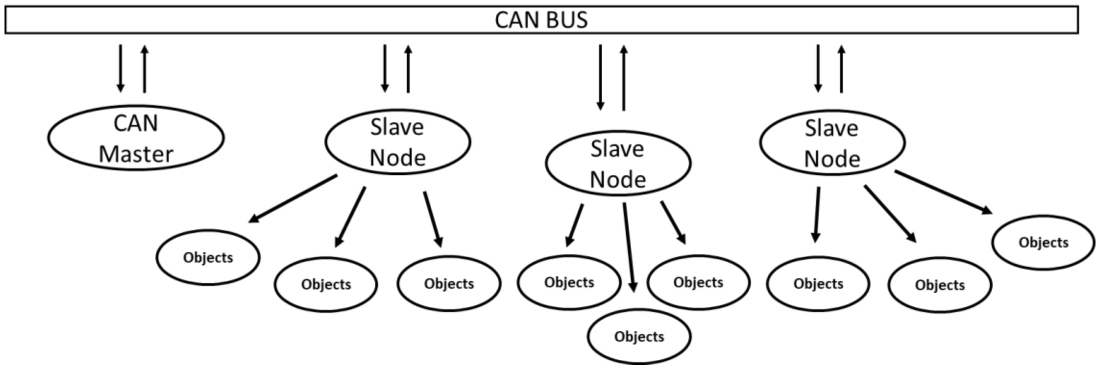


Figure 3-1: CAN Network

3.2 CANopen Message Structure

CAN messages or telegrams are used to exchange data between the host (master) and a node (slave). Table 3-2: CAN Telegram shows an outline of the standard and extended formats of a CAN message.

Table 3-2: CAN Telegram

Start Field	Arbitration Field	Control Field	Data Field	CRC Field	ACK Field	End Field
1 bit	12 or 32 bits	6 bits	0-64 bits	16 bits	2 bits	7 bits

The extended format has a 29- bit identifier in the Arbitration Field- which has the benefit of increasing the number of types of messages that can be supported on the CAN network from 2032 to 500 million. The penalty of increasing the length of the telegram is bandwidth. For example, in the case of cyclic data, the maximum net transfer rate of a standard frame is 577kbit/s while the extended frame is 488kbit/s. Due to the architecture of the telegram both types can be used on a single CAN network. The Glentek Drive uses the standard format to transmit and receive CAN telegrams.

The CANopen protocol uses CAN telegrams to get data and configure slave behavior by modifying the Arbitration, Control, and Data Field (Table 3-3: CANopen Message Using a Standard CAN Telegram). We refer to these telegrams as CANopen messages.

Table 3-3: CANopen Message Using a Standard CAN Telegram

Start Field	Arbitration Field		Control Field			Data Field
Start Bit	COB-ID	RTR	IDE	Reserved	Data Length Code	Data
1 bit	11 or 31 bits	1 bit	1 bit	1 bit	4 bits	0-64 bits

In the case of a Standard CAN message, the Arbitration field consists of two sub fields- the COB-ID or 11- bit identifier and the RTR – single remote transmission request. These two fields establish the priority of the CANopen message. Priority arbitration is fulfilled by the CAN controller chips of each device on the CAN Bus at the hardware level. In addition to establishing priority, the COB-ID defines the type of CANopen message as well as the corresponding Node-ID that the message pertains to (Table 3-4: COB-ID).

Table 3-4: COB-ID

COB-ID	
CANopen Message Specifier	Node-ID
4 bits	7 bits

There are 7 types of CANopen messages that the arbitration field can specify- corresponding to the top 4 bits allotted for CANopen types in the COB-ID (Table 3-6: CANopen Message Types and COB-IDs). The remaining 7 bits of the COB-ID specify the unique node-id of a slave on the network. It is easy to identify the type of CANopen messages based on the COB-ID. For example, a CANopen message that has COB-ID = 583h corresponds to an SDO send (580h) from the master to a slave with unique Node-ID = 3. The lower 7 bit of the COB-ID imply the range of valid Node-IDs are from 1-127 (Node-ID = 0 corresponds to the CAN Master).

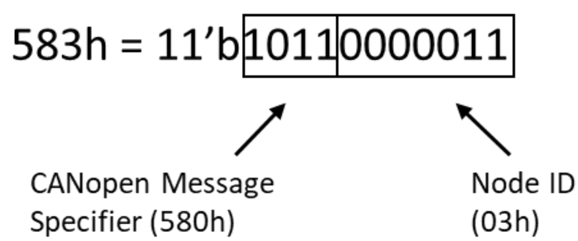


Figure 3-2: COB-ID Identification Example

The RTR bit in the Arbitration field of (Table 3-3: CANopen Message Using a Standard CAN Telegram) is relevant in the context of heartbeat monitoring and node guarding- which can be used to inform the CAN bus the state of a drive in real- time.

The data field in a CAN telegram consists of 0-64 bits (0- 8 bytes) (Table 3-5: Data Field in CANopen Messages). The contents as well as the size of the data in the data field depend on the type of the CANopen message. For example, an NMT request requires two bytes of data. But an SDO write request requires 8 bytes in the data field. The Data Length Code in the Control Field (Table 3-3: CANopen Message Using a Standard CAN Telegram) informs the CAN bus the length of data in the corresponding CANopen message. In addition, data in this field is transmitted in little endian format. For example, a 4-byte telegram with data=12345678h must be converted to LSB (Least Significant Byte first) as 78563412h prior to sending over the CAN bus.

Table 3-5: Data Field in CANopen Messages

Data Field						
D0	D1	D2	D3	D4	D5	D6
8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits

Table 3-6: CANopen Message Types and COB-IDs lists all the CANopen messages along with their COB-IDs in terms of descending priority. The full standard for CANopen messages is given in CiA 301: CANopen Application Layer and Communication Profile. Here the (TX) and (RX) are from the perspective of the slave node.

Table 3-6: CANopen Message Types and COB-IDs

Message Type	COB-ID (hex)	Description
NMT	0h	Network Management Telegram – sent from Master to control the network state of the slave.
SYNC	80h	Synchronization Message – signals to slaves to synchronize process data exchange (PDOs).
EMCY	80h +Node ID	Emergency Message – sent by slave nodes to indicate an error condition.
TIME	100h	Time Stamp
PDO 1 (TX)	180h + Node-ID	Process Data Object – sent Master and Slave to transfer process data. (If PDO1 to PDO4 are used, the Nod-ID can be assigned 1 to 127. If PDO1 to PDO8 are used, the Nod-ID can be assigned 1 to 63.)
PDO 1 (RX)	200h + Node-ID	
PDO 2 (TX)	280h + Node-ID	
PDO 2 (RX)	300h + Node-ID	
PDO 3 (TX)	380h + Node-ID	
PDO 3 (RX)	400h + Node-ID	
PDO 4 (TX)	480h + Node-ID	
PDO 4 (RX)	500h + Node-ID	
PDO 5 (TX)	680h + Node-ID	
PDO 5 (RX)	780h + Node-ID	
PDO 6 (TX)	1C0h + Node-ID	
PDO 6 (RX)	240h + Node-ID	
PDO 7 (TX)	2C0h + Node-ID	
PDO 7 (RX)	340h + Node-ID	
PDO 8 (TX)	3C0h + Node-ID	
PDO 8 (RX)	440h + Node-ID	
SDO (TX)	580h + Node-ID	Service Data Object – initiated by Master to exchange configuration data in slave.
SDO (RX)	600h + Node-ID	
NMT Error Control	700h + Node-ID	NMT Error Control – Used in Node/Life Guarding and Heartbeat Protocols.

4. CANopen Messages

4.1 Network Management (NMT)

Network management telegrams (NMT) are sent from the master to slave nodes in order to control the network state of the slaves. Immediately after a power cycle or a hardware reset, the NMT state is Boot Up (Initialization). In the Boot Up state, the Drive is initializing and will not respond to any CANopen messages it receives. After the booting process has completed, the NMT state changes automatically to the Pre-Operational state. In Pre-Operational state, the drive can respond to NMT and SDO CANopen messages; however, it will not respond to PDOs. In Operational state, the drive can respond to NMT, SDOs, and PDO CANopen messages. In the Stopped state, the Drive can respond only to NMT messages. The state diagram of the Glentek Drive is presented in Figure 4-1: NMT State Machine. It shows the NMT transitions that are possible. In addition, Table 4-1: NMT State Description gives a description of the NMT states mentioned above.

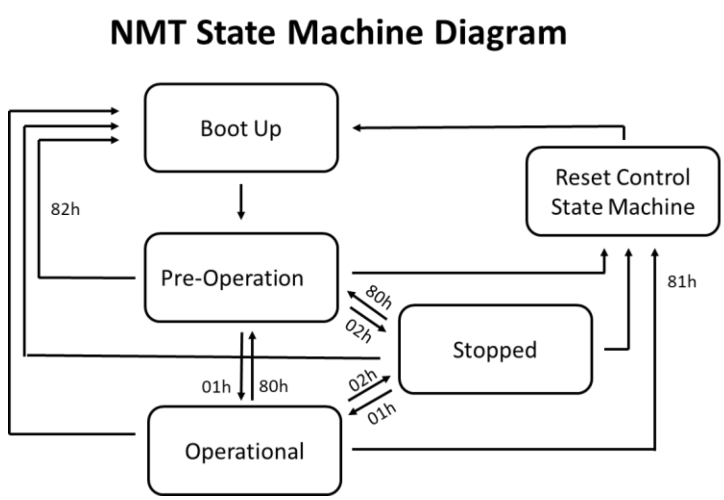


Figure 4-1: NMT State Machine

Table 4-1: NMT State Description

NMT State	Description
Boot Up	This is the state after a power cycle, or a communication reset. During this state, the drive is initializing and no communication via CAN is available. After internal initialization, the drive enters Pre- Operational automatically.
Pre- Operational	Used generally for configuration via SDOs prior to bringing the drive to Operational. All CAN message types besides PDOs are allowed in this state.
Operational	The drive is fully functional. All CAN message types can be sent.
Stopped	All CANopen message types are disabled except for NMT and Error Control messages (Heartbeat Protocol allowed if active).

Table 4-2: NMT States and CANopen Messages

Message Type	Pre-Operational	Operational	Stopped
PDO	Disabled	Allowed	Disabled
SDO	Allowed	Allowed	Disabled
SYNC	Allowed	Allowed	Disabled
TIME	Allowed	Allowed	Disabled
EMERGENCY	Allowed	Allowed	Disabled
NMT	Allowed	Allowed	Allowed
NMT Error Control	Allowed	Allowed	Allowed

Table 4-2: NMT States and CANopen Messages shows which CANopen Messages are allowed in each NMT state.

4.1.1 NMT Messages

NMT messages allow the CAN Master to set the NMT States of nodes on the CAN bus. Among CANopen telegrams, NMT messages have the highest priority since they are default COB-ID is 00h (smaller values in the identifier field have more priority). These messages are always initiated by the CAN Master. NMT messages require 2 bytes of data. The format of an NMT message is illustrated in Table 4-3: NMT CANopen Message.

Table 4-3: NMT CANopen Message

COB-ID	RTR	D0	D1	D2	D3	D4	D5	D6	D7
0	0	Byte 0	Byte 1	Not Sent					

They consist of the COB-ID and two data bytes; where byte D0 represents the command from the master and byte D1 represents the node the message is directed to. Table 4-4: NMT Message Byte 0 shows a list of the NMT commands supported by the Glentek Drive. Table 4-5: NMT Message Byte 1 shows how byte D1 is associated to the targeted Node-ID on the CAN bus.

Table 4-4: NMT Message Byte 0

Byte 0	NMT Request
1h	Sets the CANopen Slave to NMT Operational State
2h	Sets the CANopen Slave to NMT Stopped State
80h	Sets the CANopen Slave to NMT Pre-Operational State
81h	Sets the CANopen Slave to Boot Up State
82h	Sets the CANopen Slave to Boot Up State and clears Communication objects (object dictionary entries in the range 1xxxh)

Table 4-5: NMT Message Byte 1

Byte 1	Device to Address
0h	All Devices on the CAN Network
1h-7Fh	Node ID

4.2 SDO Messages

SDO messages are acyclic data transfers used to read from or write to object dictionary entries. These messages have low priority since their COB-IDs (601h and 581h) are relatively large. Their primary use is for the configuration of object dictionary information in slave nodes. There are several kinds of SDO messages, but Glentek Drives support Expedited SDO Messages exclusively. Expedited SDO messages are used when the Object Data type is 4 bytes or less. In the case that the Object Data is more than 4 bytes, Segmented SDO messages are required. Regardless of type of SDO, all SDO messages are initiated by the CAN Master and require a confirmation message from a Slave Node. The Data Field for these telegrams are outlined in Table 4-6: Expedited SDO Messages.

4.2.1 Expedited SDO Messages

These messages are used in data transfers when the object data type is less than or equal to 4 bytes (i.e. the corresponding object has data type: Unsigned 8-bit, Integer 8-bit, Unsigned 16-bit, Integer 16-bit, Unsigned 32-bit, and Integer 32-bit). All expedited SDO messages involve two CANopen messages: a request from the Master and a confirmation from the slave. In addition, there are two kinds of expedited SDO messages: Download (Write) Requests and Upload (Read) Requests. In the case of Download Requests, the CAN Master seeks to write data into an object dictionary entry in a slave node. In the case of an Upload Request, the CAN Master is querying a slave node for data in its object dictionary. After each request, the CAN node will respond with a confirmation or an error message in the case an SDO request failed. The Command Byte pertaining to read requests are given in Table 4-7: Command Byte Pertaining to SDO Reads. The Command Byte pertaining to write requests are given in Table 4-8: Command Byte Pertaining to SDO Write.

Table 4-6: Expedited SDO Messages

Arbitration Field		Data Field							
COB-ID	RTR	D0	D1	D2	D3	D4	D5	D6	D7
580h + Node-ID	0	Command Byte	Object Index		Sub-Index	Object Data			
600h + Node-ID	0	Command Byte	Object Index		Sub-Index	Object Data			

Table 4-7: Command Byte Pertaining to SDO Reads

Command Byte	Description
40h	Read Request. Initiated by the master to request a slave node to upload data.
41h	Segmented Read Confirmation. Used by the slave node to indicate
42h	Expedited Read Confirmation. Used by slave node to indicate a data transfer of 4 bytes or less.
4Fh	Expedited Read Confirmation. Used by slave node to indicate a data transfer of 1 byte.
4Bh	Expedited Read Confirmation. Used by slave node to indicate a data transfer of 2 bytes.
43h	Expedited Read Confirmation. Used by slave node to indicate a data transfer of 4 bytes.
60h	Segmented request from the host. The CAN master toggles between 60h and 70h each time the slave node responds with data.
70h	
00h	Response from slave node indicating that previous segmented request had Command Byte 60h, but should have been 70h instead.
01h	Response from slave node indicating corresponding CAN Message has the last segmented data. Occurs if the last Segmented Request from the CAN Master was 60h.
10h	Response from slave node indicating to the CAN Master that previous segmented request had Command Byte 70h, but should have been 60h instead.
11h	Response from slave node indicating corresponding CAN message has the last segmented data. Occurs if the last segmented request from the CAN Master was 70h.

Table 4-8: Command Byte Pertaining to SDO Write

Command Byte	Description
20h	Segmented Write Confirmation. Used by slave node to indicate a data transfer requires more than 4 bytes of data. The number of bytes to transfer are not indicated.
21h	Segmented Write Request. The number of bytes to transfer are not indicated.
2Fh	Expedited Write Request of 1 byte. Data is located at D5.
2Bh	Expedited Write Request of 2 bytes. Data is located at D5-D6.
23h	Expedited Write Request of 4 bytes. Data is located at D5-D8.
60h	Expedited and first Segmented Write Response from slave node.
20h	Segmented response from the slave node. Initially, the slave node responds with 60h, and then subsequently toggles between 20h and 30h.
30h	
00h	Segmented Write request if the previous slave node response is 60h or 30h.
01h	Segmented Write Request if the previous slave node response is 60h or 30h and the corresponding message contains the last data to transfer.
10h	Segmented Write Request if the previous Slave node response ix 20h.
11h	Segmented Write Request if the previous slave node response is 20h and the corresponding message contains the last data to transfer.

4.2.2 Segmented SDO Messages

Segmented SDOs are used to access objects whose data type is greater than 4 bytes (i.e. a STRING). Like all SDOs, a segmented SDO is initiated by the master. The initial transaction establishes the number of bytes to transfer. In the case of a Download (Write) Request, the Object Data of the first SDO from the Master is the number of bytes to transfer. In the case of an Upload (Read) Request, the confirmation SDO from the slave tells the master the number of bytes to transfer. Subsequent SDOs involve the actual data exchange from the object dictionary. They contain a Toggle and a Last Segment bit. During actual data transfer, the master must alternate the toggle bit. In addition, the last segment bit is set to 1 when the associated CANopen message contains the last data transfer. In addition, Segmented SDOs cannot be interrupted by expedited SDOs; only one type of SDO may be transmitted at a time. For more information on Segmented SDO Messages refer to CiA301.

4.2.3 SDO Abort Transfer Messages

If there is an error during an upload or download of an object, the corresponding slave node responds with an abort transfer message. The Command Byte returned by the slave will be 80h. Bytes D1-D3 state the Object Index and Sub-Index as usual. The Object Data in D4-D7 tells the Abort Code and ultimately reason of why the SDO transfer failed. The data field of these telegrams are outlined in Table 4-9: SDO Abort Messages.

Table 4-9: SDO Abort Messages

Arbitration Field		Data Field							
COB-ID	RTR	D0	D1	D2	D3	D4	D5	D6	D7
580h + Node-ID	0	80h	Object Index		Sub-Index	Error			

Table 4-10: SDO Abort Messages Example gives an example of an SDO abort in which the master provides the incorrect command byte for a write request. The master attempts to write to Object 607Ah: Target Position which has data type Unsigned 32-bit - with a 1 byte write request (D0 = 2Fh). The slave responds with abort code 06070010h - length of service parameter does not match. Table 4-11: Abort Codes lists the Abort Codes and their descriptions.

Table 4-10: SDO Abort Messages Example

Arbitration Field		Data Field								
COB-ID	RTR	D0	D1	D2	D3	D4	D5	D6	D7	Description
604h	0	2Fh	7Ah	60h	00h	80h	00h	00h	00h	Master sends a 1-byte SDO write request to Node ID 4. Data written to Object Index 607Ah Sub-Index 0 is 80h. Target Position configured set to 80h
584h	0	80h	7Ah	60h	00h	10h	00h	07h	06h	SDO Abort from Node ID: 4. Abort code Data: 06070010h does not match, length of service parameter does not match.

Table 4-11: Abort Codes

Abort Code	Description
05030000h	Toggle bit not alternated.
05040000h	SDO protocol timed out.
05040001h	Client/server command specifier not valid or unknown.
05040002h	Invalid block size (block mode only).
05040003h	Invalid sequence number (block mode only).
05040004h	CRC error (block mode only).
05040005h	Out of memory.
06010000h	Unsupported access to an object.
06010001h	Attempt to read a write only object.
06010002h	Attempt to write a read only object.
06020000h	Object does not exist in the object dictionary.
06040041h	Object cannot be mapped to the PDO.
06040042h	The number and length of the objects to be mapped would exceed PDO length.
06040043h	General parameter incompatibility reason.
06040047h	General internal incompatibility reason.
06060000h	Access failed due to a hardware error.
06070010h	Data type does not match, length of service parameter does not match.
06070012h	Data type does not match, length of service parameter too high
06070013h	Data type does not match, length of service parameter too low
06090011h	Sub-Index does not exist.
06090030h	Invalid value for parameter (download only).
06090031h	Value of parameter written too high (download only).
06090032h	Value of parameter written too low (download only).
06090036h	Maximum value is less than minimum value.
060A0023h	Resource not available: SDO connection
08000000h	General error
08000020h	Data cannot be transferred or stored to the application
08000021h	Data cannot be transferred or stored to the application because of local control.
08000022h	Data cannot be transferred or stored to the application because of the present device state.
08000023h	Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of an file error).
08000024h	No data available

4.2.4 SDO Message Examples

Only expedited SDO messages are covered here since Glentek Drives do not have objects with data type larger than Unsigned 32-bit. Table 4-12: Example Expedited SDO Write Request shows a write request initiated by the CAN master (all SDOs are initiated by the CAN Master) and the subsequent confirmation from the slave. The CAN master seeks to write data 000Fh to the Object: Control Word in slave node id 3. The Master constructs the SDO by using an SDO (RX) CANopen message. This is done by summing the default COB-ID for a received SDO (600h) with the node id (i.e. 600h + 03h) (Table 3-6: CANopen Message Types and COB-IDs). The RTR bit in the Arbitration field is left zero in expedited SDO messages.

Table 4-12: Example Expedited SDO Write Request

Arbitration Field		Data Field							
COB-ID	RTR	D0	D1	D2	D3	D4	D5	D6	D7
603h	0	2Bh	40h	60h	00h	0Fh	00h	00h	00h
583h	0	60h	40h	60h	00h	00h	00h	00h	00h

The data field follows the form outlined in Table 4-6: Expedited SDO Messages. D0 represents the command byte. Since the data type of Object 6040h: Controlword is Unsigned 16-bit or 2 bytes, the write request has command byte 2Bh (Table 4-13: Example Expedited SDO Read Request). The Object Index of Control Word is placed in D1- D2 least significant bit first (LSB). The Sub-Index of Control Word is placed in D3. The data requested to be written to Control Word is placed in bytes D4-D5 LSB. The remaining bytes (D6-D7) are set to zero.

In the case the download request succeeds, the expected COB-ID is the sum of SDO(TX) and the slave Node-ID, or 583h. The command byte for a successful download confirmation is 60h (Table 4-12: Example Expedited SDO Write Request). The index and Sub-Index of the object dictionary entry is unchanged; but the Object Data D4-D7 for a write response is all zeros.

Table 4-13: Example Expedited SDO Read Request

Arbitration Field		Data Field							
COB-ID	RTR	D0	D1	D2	D3	D4	D5	D6	D7
603h	0	40h	40h	60h	00h	00h	00h	00h	00h
583h	0	4Bh	40h	60h	00h	0Fh	00h	00h	00h

Table 4-13: Example Expedited SDO Read Request shows an example of an Upload Request initiated by the CAN master (all SDOs are initiated by the CAN Master) and the subsequent confirmation from the slave. The CAN master seeks to now read the Control Word from the slave with unique Node-ID=3. The COB-ID is 603h; unchanged from the previous example. The Command Byte for a read request is 40h. Similarly, the Control Word Object Index and Sub-Index are unchanged from the previous example. D4-D7 of the read request is set to all zeros.

In the case the read request was a success, the confirmation packet will be the 603h packet in Table 4-13: Example Expedited SDO Read Request. Since the data type of Control Word is Unsigned 16-bit, the Command Byte of a successful Upload Confirmation is 4Bh. Just as in the Upload Request, the Upload Confirmation has index 6040h and Sub-Index 0 in D1-D2 (LSB) and D3, respectively. And the returned data, uses D4-D5 (LSB)– again since Control Word has data type Unsigned 16-bit.

4.3 NMT Error Control Messages

NMT Error Control Messages are used to detect failures within a CAN-based network. These messages consist of the COB-ID and one data byte. There are three types of NMT Error Control Messages: Boot Up, Node/Life Guarding, and Heartbeat. Glentek Drives do not support Node/Life Guarding but do support Boot Up and Heartbeat. Table 4-14: NMT Error Control CANopen Messages shows the format of the NMT Error Control message. Table 4-15: Byte 0 in NMT Error Control Messages shows how to decode Byte 0 into the NMT states.

Table 4-14: NMT Error Control CANopen Messages

COB-ID	RTR	D0	D1	D2	D3	D4	D5	D6	D7
700h + Node ID	0	Byte 0	Not Sent						

Table 4-15: Byte 0 in NMT Error Control Messages

Byte 0	NMT State
0h	Boot Up
4h	Stopped
7Fh	Pre- Operational
5h	Operational

4.3.1 Boot Up

Table 4-16: NMT Boot Up Example shows an example of how NMT message requests from the Master and NMT Error Control from slave devices on the CAN bus behave. Initially, the CAN Master requests node 5 to go to the Boot Up state. The Slave Node 5 respond with the NMT Error Control message Boot Up. Shortly after, Node ID 5 responds with an Error Control Message stating that its NMT state has changed to Pre-Operational.

Table 4-16: NMT Boot Up Example

Arbitration Field		Data Field								
COB-ID	RTR	D0	D1	D2	D3	D4	D5	D6	D7	Description
000h	0	81h	05h	Not Sent						CAN Master requests Node-ID 5 on the bus to go to Reset Control State Machine.
705h	0	00h	Not Sent							Node-ID 5 reports a state change to Boot Up state.
705h	0	7Fh	Not Sent							Node-ID 5 reports a state change to Pre- Operational State

4.3.2 Error Control: Heartbeat

Heartbeat Protocols is an error control service used to detect failures within a CAN-based network. When the Heartbeat protocol is used, the slave node transmits an NMT Error Control message to the bus. However, instead of being polled by the CAN Master, the slave transmits its Error Protocol Message cyclically. Heartbeat protocol is configured by the CAN master using Object 1016h: Consumer Heartbeat Time and Object 1017h: Producer Heartbeat Time. These object dictionary entries represent the cycle time of the heartbeat transmission from the servo drive in units of milliseconds. Heartbeat transmission from the slave node begins once either Object 1016h: Consumer Heartbeat Time or Object 1017h: Producer Heartbeat Time are set. Slave nodes can be configured to be either Heartbeat Producers or Consumers.

Table 4-17: Heartbeat Producer shows how an example of configuring Node-ID 4 to send heartbeats at a rate of 2500 [ms]. The CAN Master sends an expedited SDO to Object 1017h: Producer Heartbeat Time Sub-Index 1 in Node-ID 4. Since the data type of Object 1017h: Producer Heartbeat Time is Unsigned 16-bit (2 Bytes), the appropriate Command Byte of the Write Request is 2Bh. The Slave responds that the SDO write request is successful. Now, configured as a heartbeat producer, Slave Node 4 sends out NMT Error Control messages every 2500 [ms].

Table 4-17: Heartbeat Producer

Arbitration Field		Data Field								
COB-ID	RTR	D0	D1	D2	D3	D4	D5	D6	D7	Description
604h	0	2Bh	17h	10h	00h	C4h	09h	00h	00h	Master sends a 2-byte SDO write request to Node-ID 4. Data written to Object Index 1017h Sub-Index 0 is 000009C4h. Heartbeat time configured to 9C4h [ms] = 2500 [ms]. Monitor node ID: 4.
584h	0	60h	17h	10h	00h	00h	00h	00h	00h	SDO write response from Node-ID: 4. Write request to Object Index 1017h, Sub-Index 0 is successful.
704h	0	7Fh	Not Sent							Node 4 sends out a heartbeat since it is a Heartbeat Producer.
704h	0	7Fh	Not Sent							Node 4 sends out a heartbeat since it is a Heartbeat Producer.

Table 4-18: Heartbeat Producer/Consumer Example

Arbitration Field		Data Field								
COB-ID	RTR	D0	D1	D2	D3	D4	D5	D6	D7	Description
604h	0	2Bh	17h	10h	00h	C4h	09h	00h	00h	Master sends a 2-byte SDO write request to Node-ID 4. Data written to Object Index 1017h, Sub-Index 0 is 9C4h. Heartbeat time configured to 9C4h [ms] = 2500 [ms].
584h	0	60h	17h	10h	00h	00h	00h	00h	00h	SDO Write Response from Node-ID 4. Write request to Object Index: 1017h, Sub-Index 0 is successful.
603h	0	23h	16h	10h	01h	B8h	0Bh	00h	00h	Master sends a 4-byte SDO write request to Node-ID 3. Data written to Object Index 1016h Sub-Index 1 is 00000BB8h. Heartbeat time configured to BB8h [ms] = 3000 [ms]. Monitor Node-ID: 4.
583h	0	60h	16h	10h	01h	00h	00h	00h	00h	SDO write response from Node-ID 3. Write request to Object Index 1016h Sub-Index 1 is successful.
704h	0	7Fh	Not Sent							Node-ID 4 sends out a heartbeat since it is a Heartbeat Producer.
703h	0	7Fh	Not Sent							Node-ID 3 responds with NMT Error Control since it is a Heartbeat Consumer.

Table 4-18: Heartbeat Producer/Consumer Example shows an example of configuring slave node 4 to be a heartbeat producer and slave node 3 to be a heartbeat consumer. Initially, the CAN Master sends a 2-byte expedited SDO to slave Node-ID 4 by setting the Object 1017h: Producer Heartbeat Time. Later, the CAN Master sends a 3-byte expedited SDO to slave Node-ID 3 by setting the Object 1016h: Consumer Heartbeat Time to 3000 [ms]. After this Node-ID 4 transmits NMT Error Control Messages every 2500 [ms]. Node-ID 3 – the heartbeat consumer guards the reception of the heartbeat within Object 1016h: Consumer Heartbeat Time. It is important to note that Object 1016h: Consumer Heartbeat Time should always be greater than Object 1017h: Producer Heartbeat Time. If the heartbeat is not received within the heartbeat consumer time, a heartbeat event will be generated.

4.4 SYNC Message

A SYNC message is a high priority broadcast message sent by a sync producer (usually the CAN Master but slaves can be sync producers as well) to all the slave nodes on the bus. In the case that a slave node has PDOs configured to be synchronous, the slave will send/process its TX/RX PDO data based on the sync. Ultimately, the sync message allows all the nodes on a CAN bus to execute and sample their data at the same time. A sync message consists of COB-ID only; there are no data bytes. Refer to Table 4-19: Sync CANopen Messages for the form of a Sync message.

Table 4-19: Sync CANopen Messages

Arbitration Field		Data Field							
COB-ID	RTR	D0	D1	D2	D3	D4	D5	D6	D7
80h	0	Not Sent							

4.5 EMERGENCY Messages

Emergency messages are sent to indicate to the bus an error condition in a slave node. They are transmitted only once per error event. Table 4-20: Emergency CANopen Message outlines the format of an emergency message. These messages are 8 data bytes long. D0 and D1 indicate the EMCY Code as dictated by CiA301. D2 indicates the error register. The error register is the same 8-bit value as Object 1001h- Error Register. The remaining bytes D3-D7 are Glentek specific error codes. A list of Glentek specific error codes is listed in the section EMERGENCY Error Codes.

Table 4-20: Emergency CANopen Message

Arbitration Field		Data Field							
COB-ID	RTR	D0	D1	D2	D3	D4	D5	D6	D7
80h + Node-ID	0	Emergency Code		Error Register	Glentek Error Code				

4.5.1 EMERGENCY Error Codes

The bitwise fields in the Emergency Message can be interpreted by the tables below. Note that D2 the Error Register is also mirrored in object: Object 1001h: Error Register.

Table 4-21: Emergency Code List

Error Code	Description
0000h	Error Reset or No Error
1000h	Generic Error
6300h	Generic Data Set Error
8210h	PDO not processed due to length error
FF00h	Device Specific Generic Error
FF02h	State Machine State Error

Table 4-22: Error Register Code List

Error Register	Description
0	Generic Error
1	Current
2	Voltage
3	Temperature
4	Communication Error
5	Device Profile Specific
6	Reserved
7	Glentek- Specific

Table 4-23: Glentek Error Codes

Error Code	Description
64h	Out of Range
65h	Not Ready
66h	Not Supported

4.5.2 EMERGENCY Message Examples

Table 4-24: Emergency CANopen Message shows an occurrence of an Emergency CANopen Message in the instance an inappropriate value is written via an expedited SDO. Initially, the CAN master requests Node-ID 3 to go to Pre-Operational State. The Slave responds with the error protocol message reporting the new NMT state change. The CAN Master then initiates an expedited SDO to write data 0Fh to Object 60C2h: Interpolation Time Period Sub-Index 2). Writing data to the Interpolation Time Index succeeds, so the slave responds with a Successful Write Response. Despite the write process succeeding, the value 0Fh is not valid since only values of -3, -2, -1, or 0 can be written to Interpolation Time Index. As a result, the slave issues an Emergency Message with Emergency Code: 6300h, Error Register: 0Fh, and Glentek Error Code: 64h to inform the CAN Master of the Error.

Table 4-24: Emergency CANopen Message

Arbitration Field		**Data Field**								
COB-ID	RTR	D0	D1	D2	D3	D4	D5	D6	D7	Description
000h	0	80h	03h	Not Sent						Master sends an NMT request for Node-ID 3 to go to Pre-Operational
703h	0	7Fh	Not Sent							Slave Node-ID 3 responds with NMT Error Control Message
603h	0	2Fh	C2h	60h	02h	0Fh	00h	00h	00h	Expedited SDO Write Request with data: 0Fh to 60C2h Sub-Index 2. Set Time Index to 0Fh.
83h	0	00h	63h	00h	64h	0Fh	E8h	01h	00h	Emergency Message from Slave Node.
583h	0	60h	C2h	60h	02h	00h	00h	00h	00h	Successful Write Response to 60C2h Sub-Index 2.

4.6 TIME STAMP Messages

The TIME Stamp can provide a global clock for all the nodes on the CAN bus. The COB-ID is 100h. The data field uses 6 bytes. The first 4 bytes represent the number of milliseconds after midnight the next two bytes represent the number of days since January 1, 1984.

Table 4-25: Time Stamp CANopen Message

Arbitration Field		**Data Field**							
COB-ID	RTR	D0	D1	D2	D3	D4	D5	D6	D7
100h	0	Time after midnight in milliseconds				Current Day Since 01/01/84		00h	00h

4.7 PDO Messages

PDOs (Process Data Objects) are high priority messages with minimal protocol overhead. These messages can be sent or received in NMT Operational State only. PDOs can transmit 8 bytes of data at a time. There are two kinds of PDOs- transmit and RPDOs distinguished as Transmit-PDO (TPDO) and Receive-PDO (RPDO). CANopen devices supporting TPDO are PDO producers and CANopen devices supporting RPDO are called PDO consumers. The Glentek Drive supports 8 TPDOs and 8 RPDOs. Each PDO has associated with it a Configuration and a Mapping Object that must be configured. These objects are written to via SDOs. The configuration Object specifies the transmission mode, COB-ID, and other parameters for configuring the PDO. The mapping object defines the data to be transmitted by the PDO. In addition, Glentek Drives support three Transmission Types: Event, Time, and Synchronously triggered.

4.7.1 Communication Parameter Object

The Communication Objects (1800h-1807h for TPDOs and 1400h-1407h for RPDOs) specifies the unique COB-ID, Transmission, Inhibit Time, and Event Timer. The zeroth Sub-Index “Number of Entries” has read only access and specifies the number of Sub-Indices included in the object. Table 4-27: PDO Communication Parameter 1 Sub-Index Description – provides a description of the Sub-Indices listed in Table 4-26: Object 1400h: RPDO1 Communication Parameter.

4.7.1.1 COB-ID

Sub-Index 1 enables/disables the PDO as well as specifies the COB-ID of the PDO. The default COB-IDs for RPDOs are 200h + Node-ID, 300h + Node-ID, 400h + Node-ID, and 500h + Node-ID. The default COB-IDs for TPDOs are 180h + Node-ID, 280h + Node-ID, 380h + Node-ID, and 480h + Node-ID. To enable, the PDO the top bit (bit 31) of COB-ID should be zero. To disable, the PDO bit 31 of the COB-ID should be set to one. Note that for TPDOs, RTR is not supported, therefore bit 30 must be set to 1.

Table 4-26: Object 1400h: RPDO1 Communication Parameters

Sub-Index	Description	Default	PDO Map	Data Type	Access
0	Number of Entries	05h	No	Unsigned 8-bit	RO
1	COB-ID	200h + Node-ID	No	Unsigned 32-bit	RW
2	Transmission Type	FFh	No	Unsigned 8-bit	RW
3	Inhibit Time	0000h	No	Unsigned 16-bit	RW
4	Reserved	00h	No	Unsigned 8-bit	RW
5	Event Timer	0000h	No	Unsigned 16-bit	RW

4.7.1.2 Transmission Type

Sub-Index 2 sets the Transmission Type of the PDO. Values of 1-240 in this field specify the PDO to be synchronous. In the case that Transmission Type is set to 1 for example, the slave will send/receive PDOs each time a SYNC message is received. However, if set to 4, the slave node will send/receive the corresponding PDO after receiving 4 SYNC messages. Values of 254 or 255 in Transmission Type specify the PDO to be asynchronous. In the case of 254, the PDO transmits in the instance an event timer expires. If set to 255, the PDO transmits in the case that either an event timer expires, or an object mapped in the PDO changes value.

4.7.1.3 Inhibit Type

Sub-Index 3 Inhibit Time specifies the minimum time interval to transmit a PDO. This value is defined in multiples of 100 [us]. The value 0 disables the inhibit time.

4.7.1.4 Reserved

Sub-Index 4 is reserved.

4.7.1.5 Event Timer

Sub-Index 5 is a two-byte word that contains the Event Timer- relevant to Transmission Type 254 or 255. The Event Timer value defines the maximum interval for TPDO transmission in milliseconds. If the internal timer exceeds Event Timer, the corresponding PDO is transmitted, and the internal timer is reset.

Table 4-27: PDO Communication Parameter 1 Sub-Index Description

Sub-index	Description
0	Number of Entries: The number of Sub-Indices in the object.
1	COB-ID: Sets the COB-ID of the corresponding PDO. Note that for TPDOs, RTR is not supported, therefore bit 30 must be set to 1.
2	Transmission Type: Value 0: N/A Value 1-240: Sets the rate data is accepted on sync packet. For example, the value 2 specifies that a PDO is accepted every other sync packet. Value 241-251: Reserved Value 252, 253: Not Supported Value 254: Transmits the PDO at the rate of the event timer. Value 255: Transmit PDO at the rate of the event timer or if a value of an object mapped into the PDO has changed.
3	Inhibit Time: Sets the minimum time between PDO transmission. Minimum Time = Value*100 microseconds.
4	Compatibility Entry: This field is reserved and should be set to zero.
5	Event Timer: Sets the maximum time (in milliseconds) between transmissions of the PDO. Used in the case Transmission type is set to 254 or 255.

4.7.2 Mapping Parameter Objects

Table 4-28: Object 1600h: RPDO Mapping Parameter

Sub-Index	Description	Default	PDO Map	Data Type	Access
0	Number of Entries	01h	No	Unsigned 8-bit	RW
1	Mapping Entry 1	60410010h	No	Unsigned 32-bit	RW
2	Mapping Entry 2	00000000h	No	Unsigned 32-bit	RW
3	Mapping Entry 3	00000000h	No	Unsigned 32-bit	RW
4	Mapping Entry 4	00000000h	No	Unsigned 32-bit	RW
5	Mapping Entry 5	00000000h	No	Unsigned 32-bit	RW
6	Mapping Entry 6	00000000h	No	Unsigned 32-bit	RW
7	Mapping Entry 7	00000000h	No	Unsigned 32-bit	RW
8	Mapping Entry 8	00000000h	No	Unsigned 32-bit	RW

The Mapping Object (1A00h-1A07h for TPDOs and 1600h-1607h for RPDOs) specifies the number of objects in a PDO as well as the mapped objects in the PDO. Sub-Index 0 “Number of Entries” has read/write access and specifies the number of Sub-Indices included in the object. The remaining 8 Sub-Indices are Mapping Entry 1-8. Each of these Sub-indexes are 4 bytes wide and used to map an object dictionary entry to the corresponding PDO. For each Mapping Entry, bits 0-7 specify the length of the object. Bits 8-15 specify the Sub-Index of the object to be mapped. Bits 16-31 specify the Index of the object to be mapped (Refer to Table 4-30: Mapping Entry). Note that PDOs can transmit a maximum of 8 bytes of data.

Table 4-29: PDO Mapping Parameter Sub-Index Description

<table>
<tr><th>Sub-Index</th><th>Description</th></tr>
<tr><td>0</td><td>Number of Entries: The number of mapped Sub-Indices in the object.</td></tr>
<tr><td>1</td><td rowspan="8">Mapping Entries: Set the objects to be mapped to the corresponding PDO. The maximum size of data that a PDO can receive/transmit is 8 bytes.</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td></tr>
<tr><td>4</td></tr>
<tr><td>5</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td></tr>
<tr><td>8</td></tr>
</table>

Therefore, the arithmetic sum of the data types of each of the mapping entries cannot exceed 8 bytes. For example if Object 6064h: Position Actual Value is assigned to Mapping Entry 1 and Object 6065h: Following Error Window is assigned to Mapping Entry 2, no more objects can be mapped to that PDO since the data type of both Position Actual Value and Following Error Window are Unsigned 32-bit or 4 bytes.

Table 4-30: Mapping Entry Object Data

Data Field			
D4	D5	D6	D7
Object Index		Sub-Index	Length

4.7.3 PDO Configuration

RPDOs have Configuration Objects 1400h-1407h and 1600h-1607h. TPDOs have Configuration Objects 1800h-1807h and 1A00h-1A07h. The Objects: Object 1400h: RPDO1 Communication Parameter and Object 1600h: RPDO1 Mapping Parameter are used to configure RPDO 1. Similarly, Object 1401h: RPDO2 Communication Parameter and Object 1601h: RPDO2 Mapping Parameter are used to configure RPDO2 and so on for the remaining RPDOs and TPDOs. The user must first identify which PDO they want to use.

To configure a PDO, first make sure the drive is in NMT- Preoperational Mode. The next step is to define the COB-ID of the PDO in the Communication Parameter Object- Sub-Index 1. Bit 31 of the COB-ID disables the PDO (set to 1) or enables the PDO (set to 0). Set the appropriate COB-ID and bit 31 so that the PDO is disabled. Set the “Number of Entries” in the Communication Object to zero. We then can add the mapping entries that the 8 bytes of PDO data will map. For example, if Mapping Entry 1 and 2 both have length four, the first four bytes of PDO data will represent the object in Mapping Entry 1 and the remaining bytes will represent Mapping Entry 2. After setting the Mapping Entries in the PDO Mapping Parameter Object, set the Number of Entries to reflect the number of Mapping Entries used. The next step is to enable the PDO by setting bit 31 of the COB-ID in the PDO Communication Parameter Object, Sub- Index 1 to zero.

4.7.4 PDO Configuration Example

In this example we assign Object 6064h: Position Actual Value and Object 6041h: Statusword to TPDO1 (180h + Node-ID). Since Object 6064h: Position Actual Value has data type Unsigned 32-bit and Object 6041h: Statusword has data type Unsigned 16-bit, 180h + Node-ID is a PDO with 6 bytes of data; the first four bytes will correspond to Object 6064h: Position Actual Value and the next two bytes will correspond to the Object 6041h: Statusword. In addition, we assign Object 607Ah: Target Position and Object 6040h: Controlword to RPDO1 (200h + Node-ID). Since the data types for Object 607Ah: Target Position and Object 6040h: Controlword are Unsigned 32-bit and Unsigned 16-bit, respectively, 200h + Node-ID is also a PDO with 6 bytes of data. The first 4 bytes of the RPDO will correspond to the Object 607Ah: Target Position, the next 2 bytes to Object 6040h: Controlword. Recall, 8 bytes is the maximum amount of data that a PDO can transmit because CANopen messages and CAN telegrams in general have only 8 bytes allotted for their data.

1. The drive must first be in NMT Pre-Operational Mode.
2. Disable the PDO and assign COB-IDs based on the Node-ID of the drive. Bit 31 of Sub-Index 1 in Object 1800h: TPDO1 Communication Parameter and Object 1400h: RPDO1 Communication Parameter are used to disable the corresponding PDO.
   If the unique Node-ID =1 for example, set Object 1800h: TPDO1 Communication Parameter Sub-Index 1 to C0000181h. Similarly, set the top bit in Sub-Index 1 of Object 1400h: RPDO1 Communication Parameter to disable the PDO as well as assign the COB-ID by writing: 80000201h. Later, when these PDOs are enabled, their COB-IDs will be 183h and 203h, respectively.
3. Set the number of entries in each PDO to zero. TPDO1 has Object 1A00h: TPDO1 Mapping Parameter. Set the Number of Entries in TPDOs (Sub-Index 0) to 0. Similarly, RPDO1 has Object 1600h: RPDO1 Mapping Parameter. Set the Number of Entries in Object 1600h: RPDO1 Mapping Parameter (Sub-Index 0) to 0.
4. Now that the PDOs are reset and disabled, we assign the objects to be mapped by the PDO into the Mapping Objects (Object 1A00h: TPDO1 Mapping Parameter and Object 1600h: RPDO1 Mapping Parameter). The non-zero entries in the TPDO/RPDO Mapping Objects use a 4-byte value that encodes the following information: Object Index, Sub-Index, and the length (the data type of the object) of the data into D4-D5, D6,D7, respectively (Table 4-30: Mapping Entry Object Data).

   For example, Object 607Ah: Target Position has Sub-Index 0 and length: 4 bytes or 32 bits(20h). As a result, we set Sub-Index 1 in Object 1A00h: TPDO1 Mapping Parameter to 607A0020h.

For the TPDO we set in Object 1A00h: TPDO1 Mapping Parameter:
Sub-Index 1: 60640020h
Sub-Index 2: 60410010h

For the RPDO we set in Object 1600h: RPDO1 Mapping Parameter:
Sub-Index 1: 607A0020h
Sub-Index 2: 60400010h

5. Now we set the Number of entries in the PDO Mapping Objects accordingly. The TPDO has object Target Position and Control Word, so the Number of Entries in Object 1A00h: TPDO1 Mapping Parameter (Sub-Index 0) should be set to 2. Similarly, the Number of Entries in Object 1600h: RPDO1 Mapping Parameter (Sub-Index 0) should be set to 2.
6. Set the Transmission type of Object 1800h: TPDO1 Communication Parameter and Object 1400h: RPDO1 Communication Parameter (Sub-Index 2) to Sync to 1. This means that for every Sync message received, the drive will expect an RPDO as well as send its own TPDO. Appropriate values for Sync vary from 1-240. Note, that in the case of using an event timer, a value of 254-255 is written to Sub-Index 2 instead. In addition, the Event Time, in milliseconds should be set in the Communication Object, Sub-Index 5.
7. Enable the PDO by clearing bit 31 of the COB-ID in Communication Objects (Object 1800h: TPDO1 Communication Parameter and Object 1400h: RPDO1 Communication Parameter). In the case of the transmit object, set Object 1800h: TPDO1 Communication Parameter Sub-Index 1 to 40000181h (Note bit 30 is set to 1 since RTR is disabled for TPDOs). Similarly, for the Receive Communication object, set Object 1400h: RPDO1 Communication Parameter Sub-Index 1 to 00000201h.
8. Set the drive to Operational State by issuing an NMT Message.
9. Send Sync Message and TPDO to Slave node 1.

The default COB-IDs for RX/TX PDOs are listed in Table 3-6: CANopen Message Types and COB-IDs however, the range of valid values for PDO COB-IDs is 181h – 57Fh. When setting the COB-ID, special care must be taken for choosing a unique value. Since there are multiple slaves on the bus it is possible to assign the same COB-ID to multiple PDOs.

Table 4-31: PDO Configuration Example

Arbitration Field		Data Field								Description
COB-ID	RTR	D0	D1	D2	D3	D4	D5	D6	D7	
000h	0	80h	01h	Not Sent						Master sends an NMT request for Node-ID 1 to Pre-Operational State.
701h	0	7Fh	Not Sent							Slave Node-ID 1 responds with the NMT Error Control Message.
601h	0	23h	00h	18h	01h	81h	01h	00h	C0h	Expedited SDO write request with data: C0000181h to 1800h Sub-Index 1. Disables TPDO1 and sets COB-ID to 181h.
581h	0	60h	00h	18h	01h	00h	00h	00h	00h	Successful Write response to 1800h Sub-Index 1.
601h	0	23h	00h	14h	01h	01h	02h	00h	80h	Expedited SDO write request with data: 80000201h to 1400h Sub-Index 0. Disables RPDO1 and sets COB-ID to 201h.
581h	0	60h	00h	14h	01h	00h	00h	00h	00h	Successful Write response to 1400h Sub-Index 1.
601h	0	2Fh	00h	1Ah	00h	00h	00h	00h	00h	Expedited SDO write request with data: 00h to 1A00h Sub-Index 0. Set Number of Entries to 0.
581h	0	60h	00h	1Ah	00h	00h	00h	00h	00h	Successful Write response to 1A00h Sub-Index 0.
601h	0	2F	00h	16h	00h	00h	00h	00h	00h	Expedited SDO write request with data: 00h to 1600h Sub-Index 0. Sets Number of Entries to 0.
581h	0	60h	00h	16h	00h	00h	00h	00h	00h	Successful Write response to 1600h Sub-Index 0.
601h	0	23h	00h	1Ah	01h	20h	00h	64h	60h	Expedited SDO write request with data 60640020h to 1A00h Sub-Index 1. Set Mapping Entry 1 of TPDO1 to Position Actual Value (Object Index: 6064 h Sub-Index: 00h, Size: 20h).
581h	0	60h	00h	1Ah	01h	00h	00h	00h	00h	Successful Write Response to 1A00h Sub-Index 1.
601h	0	23h	00h	1Ah	02h	10h	00h	41h	60h	Expedited SDO write request with data 60410010h to 1A00h Sub-Index 2. Set Mapping Entry 2 of TPDO1 to Control Word (Object Index: 6041h Sub-Index: 00h, Size: 10h).
581h	0	60h	00h	1Ah	02h	00h	00h	00h	00h	Successful Write Response to 1A00h Sub-Index 2.

601h	0	23h	00h	16h	01h	20h	00h	7Ah	60h	Expedited SDO Write Request with data 607A0020h to 1600h Sub-Index 1. Set Mapping Entry 1 of RPDO1 to Target Position (Object Index: 607Ah, Sub-Index: 00h, Size: 20h).
581h	0	60h	00h	16h	02h	00h	00h	00h	00h	Successful Write Response to 1600h Sub-Index 1.
601	0	23h	00h	16h	02h	10h	00h	40h	60h	Expedited SDO Write Request with data 60400010h to 1600h Sub-Index 2. Set Mapping Entry 2 of RPDO1 to Control Word (Object Index: 6040h, Sub-Index: 00h, Size: 10h).
581	0	60h	00h	16h	02h	00h	00h	00h	00h	Successful Write Response to 1600h Sub-Index 2.
601h	0	2Fh	00h	1Ah	00h	02h	00h	00h	00h	Expedited SDO write request with data: 00h to 1A00h Sub-Index 0. Sets Number of Entries to 2.
581h	0	60h	00h	1Ah	00h	00h	00h	00h	00h	Successful Write response to 1A00h Sub-Index 0.
601h	0	2Fh	00h	16h	00h	02h	00h	00h	00h	Expedited SDO write request with data: 00h to 1600h Sub-Index 0. Sets Number of Entries to 2.
581h	0	60h	00h	16h	00h	00h	00h	00h	00h	Successful Write response to 1600h Sub-Index 0.
601h	0	2Fh	00h	14h	02h	01h	00h	00h	00h	Expedited SDO write request with data: 00h to 1400h Sub-Index 2. Sets the Transmission Type to 1.
581h	0	60h	00h	14h	02h	00h	00h	00h	00h	Successful Write Response to 1400h Sub-Index 2.
601h	0	2Fh	00h	18h	02h	01h	00h	00h	00h	Expedited SDO write request with data: 00h to 1800h Sub-Index 2. Sets the Transmission Type to 1.
581h	0	60h	00h	18h	02h	00h	00h	00h	00h	Successful Write Response to 1800h Sub-Index 2.
601h	0	23h	00h	18h	01h	81h	01h	00h	40h	Expedited SDO write request with data: 40000181h to 1800h Sub-Index 1. Enables TPDO1 and sets COB-ID to 181h.
581h	0	60h	00h	18h	01h	00h	00h	00h	00h	Successful Write response to 1800h Sub-Index 1.
601h	0	23h	00h	14h	01h	02h	00h	00h	00h	Expedited SDO Write Request with data: 000000201h to 1400h Sub-Index 1. Enables RPDO1 and sets COB-ID to 201h.
581h	0	60h	00h	14h	01h	00h	00h	00h	00h	Successful Write response to 1400h Sub-Index 1.
000h	0	01h	01h	Not Sent						Master sends an NMT request for Node-ID 1 to go to Operational State.
80h	0	Not Sent								CAN Master sends a Sync Message

181h	0	11h	22h	33h	44h	08h	04h	Not Sent	CAN Slave sends TPDO1. First four bytes are Position Actual Value: 44332211h and second two bytes are Status Word 0408h.
201h	0	11h	22h	33h	44h	0Fh	00h	Not Sent	CAN Master sends RPDO1. First four bytes are Target Position: 44332211h and second two bytes are Status Word 000Fh.

4.7.5 Asynchronous Transmission Events

PDOs can also be transmitted if an event timer has expired or if the value mapped into a PDO has changed. If Transmission Type (in Communication Object Sub-Index 2) is set to 254, the corresponding PDO is transmitted when the event timer expires. The event timer period is set in Event Timer (in Communication Object Sub-Index 5). A PDO can be set to transmit if an object mapped into the PDOs value changes in addition to the event timer expiring by setting Transmission Type to 255. An example of setting the PDO to transmit according to the expiration of a timer appears in Table 4-32: Asynchronous Transmission Configuration Example.

Table 4-32: Asynchronous Transmission Configuration Example

Arbitration Field		Data Field								
COB-ID	RTR	D0	D1	D2	D3	D4	D5	D6	D7	Description
000h	0	80h	01h	Not Sent						Master sends an NMT request for Node ID 1 to enter Pre-Operational State.
701h	0	7Fh	Not Sent							Slave Node ID 1 responds with the NMT Error Control Message.
601h	0	23h	01h	18h	01h	81h	02h	00h	C0h	Expedited SDO write request with data: C0000281h to 1801h Sub-Index 1. Disables TPDO2 and sets COB-ID to 281h.
581h	0	60h	01h	18h	01h	00h	00h	00h	00h	Successful Write response to 1801h Sub- Index 1.
601h	0	2Fh	01h	1Ah	00h	00h	00h	00h	00h	Expedited SDO write request with data: 00h to 1A01h Sub-Index 0. Sets Number of Entries to 0.
581h	0	60h	01h	1Ah	00h	00h	00h	00h	00h	Successful Write response to 1A01h Sub- Index 0.
601h	0	23h	01h	1Ah	01h	20h	00h	64h	60h	Expedited SDO write request with data 60640020h to 1A01h Sub-Index 1. Set Mapping Entry 1 to Position Actual Value (6064h).
581h	0	60h	01h	1Ah	01h	00h	00h	00h	00h	Successful Write response to 1A01h Sub-Index 1.
601h	0	2Fh	01h	1Ah	00h	01h	00h	00h	00h	Expedited SDO write request with data: 01h to 1A01h Sub-Index 0. Set Number of Entries to 1.
581h	0	60h	01h	1Ah	00h	00h	00h	00h	00h	Successful Write Response to 1A01h Sub-Index 0.
601h	0	2Bh	01h	18h	05h	01h	00h	00h	00h	Expedited SDO write requests with data: 0001h to 1801 Sub-Index 5. Set Event Timer to 1 [ms].
581h	0	60h	01h	18h	05h	00h	00h	00h	00h	Successful Write Response to 1801h Sub-Index 5.
601h	0	2Fh	01h	18h	02h	FFh	00h	00h	00h	Expedited SDO Write Request with data: FFh to 1801h Sub-Index 2. Set Transmission Type to Event Timer.
581h	0	60h	01h	18h	02h	00h	00h	00h	00h	Successful Write Response to 1801h Sub-Index 2.
601h	0	23h	01h	18h	01h	81h	02h	00h	40h	Expedited SDO write request with data: 40000281h to 1801h Sub-Index 1. Enable PDO 2 and set COB-ID to 283h.

581h	0	60h	01h	18h	01h	00h	00h	00h	00h	Successful Write Response to 1801h to Sub-Index 1.
000h	0	80h	01h	Not Sent						Master sends an NMT request for Node-ID 3 to go to Operational State.
701h	0	05h	Not Sent							Slave Node-ID 3 responds with NMT Error Control Service.

5. Communication Objects

5.1 Object 1000h: Device Type

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1000h	0	Device Type	00420192h	No	Unsigned 32-bit	RO	No

This object gives information about the device type and its functionality. The first 16 bits describe the application profile. The remaining 8 bits gives additional information about the device type.

Bit	Meaning
0-15 (16 bits)	Device Profile: 402 (192h)
16-23 (8 bits)	Device Type: 02h indicating single servo drive
24-31 (8 bits)	Reserved

5.2 Object 1001h: Error Register

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1001h	0	Error Register	00h	No	Unsigned 8-bit	RO	No

This object provides error information. The bit field of the object is mapped in the following:

Error Register	Description
0	Generic Error
1	Current
2	Voltage
3	Temperature
4	Communication Error
5	Device Profile Specific
6	Reserved
7	Glentek- Specific

5.3 Object 1005h: COB-ID-Sync

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1005h	0	COB-ID Sync	00000080h	No	Unsigned 32-bit	RW	No

This object specifies the COB-ID used for the Synchronization object (SYNC). In addition, this object defines if the CAN Slave generates the SYNC.

Bit	Setting
0-10	COB-ID of the Synchronization object.
11-29	Set to 0.
30	Set to 0 to be sync consumer (Receive) Set to 1 to be a sync producer (Transmit). Sync message is produced in any operation mode: Pre-Operational, Operational, and Stopped.
31	Reserved

5.4 Object 1006h: Communication Cycle Period

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1006h	0	Communication Cycle Period	00000000h	No	Unsigned 32-bit	RW	No

This object dictates the communication cycle period (SYNC interval) in units of microseconds. This object can be set to 0 to disable to the sync message transmission.

5.5 Object 1010h: Store Parameters

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1010h	00h	Number of Entries	03h	No	Unsigned 8-bit	RO	No
1010h	01h	Save All Parameters	01h	No	Unsigned 32-bit	RW*	No
1010h	02h	Save Communication Parameters	01h	No	Unsigned 32-bit	RW*	No
1010h	03h	Save Application Parameters	01h	No	Unsigned 32-bit	RW*	No

* This object is only writable when NMT State Machine is in PreOP mode.

Current parameters are saved to EEPROM when a specific command is written to the appropriate subindex. The saved parameters are used to initialize the objects during power on the drive. The specific command is the reverse of “save”, 65766173h. Other commands are not valid.

If 65766173h is written to subindex 1, all parameters are saved.

If 65766173h is written to subindex 2, the communication parameters are saved.

If 65766173h is written to subindex 3, the application parameters are saved.

After writing the command, ‘0’ is returned during saving. If saving is succeeded, ‘1’ is returned. If an incorrect command is input, SDO abort code (0x08000020) is returned. If saving is failed, SDO abort code (0x06060000) is returned.

5.6 Object 1011h: Restore Default Parameters

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1011h	00h	Number of Entries	03h	No	Unsigned 8-bit	RO	No
1011h	01h	Restore All Default Parameters	01h	No	Unsigned 32-bit	RW*	No
1011h	02h	Restore Default Communication Parameters	01h	No	Unsigned 32-bit	RW*	No
1011h	03h	Restore Default Application Parameters	01h	No	Unsigned 32-bit	RW*	No

* This object is only writable when NMT State Machine is in PreOP mode.

Default parameters are restored and saved to EEPROM when a specific command is written to the appropriate subindex. The specific command is the reverse of "load", 64616F6Ch. Other commands are not valid.

If 64616F6Ch is written to subindex 1, all default parameters are restored and saved.

If 64616F6Ch is written to subindex 2, default communication parameters are restored and saved.

If 64616F6Ch is written to subindex 3, default application parameters are restored and saved.

After writing the command, '0' is returned during saving. If saving is succeeded, '1' is returned. If an incorrect command is input, SDO abort code (0x08000020) is returned. If saving is failed, SDO abort code (0x06060000) is returned.

5.7 Object 1014h: COB-ID EMCY

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1014h	0	COB-ID EMCY	00000080h + Node-ID	No	Unsigned 32-bit	RW	No

This object indicates the configured COB-ID for the EMCY write service.

Bits	Bitwise Definition	Value	Description
0-28	11-bit or 29-bit CAN-ID	x	11-bit CAN-ID valid 29-bit CAN-ID valid
29	Frame	0 1	11-bit CAN-ID Valid 29-bit CAN-ID valid
30	Reserved	0	Always 0
31	Valid	0 1	EMCY exists/is valid EMCY does not exist /not valid

5.8 Object 1015h: Inhibit Time EMCY

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1015h	0	Inhibit Time EMCY	0000h	No	Unsigned 16-bit	RW	No

This object indicates the configured inhibit time for the EMCY message. The value is given in multiples of 100 microseconds. The value 0 disables the inhibit time.

5.9 Object 1016h: Consumer Heartbeat Time

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1016h	0	Number of Entries	01h	No	Unsigned 8-bit	RO	No
1016h	1	Consumer Heartbeat Time	00000000h	No	Unsigned 32-bit	RW	No

The Consumer Heartbeat Time indicates the expected heartbeat cycle times. Monitoring of the heartbeat producer occurs after the reception of the first heartbeat. Note: The Consumer heartbeat time should be higher than the corresponding producer heartbeat time. The Consumer Heartbeat Time value is given in multiples of 1 [ms].

Bit	Setting
0-15	Heartbeat Time
16-23	Node-ID
24-31	Reserved – 00h

If the heartbeat time is 0 or the Node-ID is 0 or greater than 7Fh the corresponding object entry shall not be used.

5.10 Object 1017h: Producer Heartbeat Time

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1017h	0	Producer Heartbeat Time	0000h	No	Unsigned 16-bit	RW	No

The producer heartbeat time shall indicate the configured cycle time of the heartbeat. The value is given in multiples of 1 [ms]. The value 0 disables the producer heartbeat.

5.11 Object 1018h: Identity Object

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1018h	0	Number of Entries	04h	No	Unsigned 8-bit	RO	No
1018h	1	Vendor ID	000002C0h	No	Unsigned 32-bit	RO	No
1018h	2	Product Code	00009616h	No	Unsigned 32-bit	RO	No
1018h	3	Revision Number	00000502h	No	Unsigned 32-bit	RO	No
1018h	4	Serial Number	12345678h	No	Unsigned 32-bit	RO	No

The Identity Object provides general identification information of the CANopen device.

Sub-index	Description
0	Number of Entries: The number of Sub-Indices in the object.
1	Vendor ID: Contains the unique value that identifies Glentek as the Vendor.
2	Product Code: Contains the unique value that identifies a specific type of CANopen devices.
3	Revision Number: Contains the major and minor revision number of the CANopen device. The first 16 bits are the Major Revision Number. The next 16 bits are the Minor Revision Number.
4	Serial Number: This field is reserved and should be set to zero.

This following table provides a bitwise mapping for Sub-Index 3: Revision Number.

Bit	Setting
0-15	Major Revision Number
16-23	Minor Revision Number

5.12 Object 1019h: Synchronous Counter Overflow Value

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1019h	0	Synchronous Counter Overflow Value	00h	No	Unsigned 8-bit	RW	No

The object indicates the configured the highest value the synchronous counter supports. If the value is greater than 1, the SYNC message shall have a data length of 1 byte. AN EMCY message with error code 8240h – unexpected SYNC data length- may be transmitted by a SYNC consumer in the case the configured data length of the SYNC message does not meet the data length of a received SYNC message.

Value	Description
0	The SYNC message will be transmitted as a CAN message of data length 0.
1	Reserved
2-240	The SYNC message shall be transmitted as a CAN message of data length 1. The first data byte contains the counter.
241-255	Reserved

The value used should be the least common multiple of all the TPDO transmission type ($1 < \text{value} <= 240$) used. This ensures that periodic SYNC events always happen in the SYNC cycles with the same counter value.

5.13 Object 1200h: SDO Server Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1200h	0	Number of Entries	02h	No	Unsigned 8-bit	RO	No
1200h	1	COB-ID Client -> Server (RX)	0600h	No	Unsigned 16-bit	RW	No
1200h	2	COB-ID Server -> Client (TX)	0580h	No	Unsigned 16-bit	RW	No

These are the COB-IDs assigned to SDO communications from the CANopen master to the Glentek Drive.

Bits	Bitwise Definition	Value	Description
0-10	11 – Bit CAN ID	x	11-bit CAN-ID of the CAN base Frame
11-28	Reserved	00000h	These bits should not be set.
29	Frame	0 1	11-bit CAN-ID valid 29-bit CAN-ID valid
30	Dyn	0 1	Value assigned statically Value assigned dynamically
31	Valid	0 1	SDO exists SDO does not exist

5.14 Object 1280h: SDO Server Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1280h	0	Number of Entries	03h	No	Unsigned 8-bit	RO	No
1280h	1	COB-ID Client -> Server (TX)	80000000h	No	Unsigned 32-bit	RW	No
1280h	2	COB-ID Server -> Client (RX)	80000000h	No	Unsigned 32-bit	RW	No
1280h	3	Node-ID of the SDO server	40h	No	Unsigned 16-bit	RW	No

These are the COB-IDs assigned to SDO communications from the CANopen master to the Glentek Drive.

Bits	Bitwise Definition	Value	Description
0-10	11 – Bit CAN ID	x	11-bit CAN-ID of the CAN base Frame
11-28	Reserved	00000h	These bits should not be set.
29	Frame	0 1	11-bit CAN-ID valid 29-bit CAN-ID valid
30	Dyn	0 1	Value assigned statically Value assigned dynamically
31	Valid	0 1	SDO exists SDO does not exist

5.15 Object 1400h: RPDO1 Communication Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1400h	0	Number of Entries	05h	No	Unsigned 8-bit	RO	Yes
1400h	1	COB-ID	00000201h	No	Unsigned 32-bit	RW	Yes
1400h	2	Transmission Type	FFh	No	Unsigned 8-bit	RW	Yes
1400h	3	Inhibit Time	0000h	No	Unsigned 16-bit	RW	Yes
1400h	4	Reserved	00h	No	Unsigned 8-bit	RW	Yes
1400h	5	Event Timer	0000h	No	Unsigned 16-bit	RW	Yes

This object contains the communication parameters for RPDO1.
Sub-Index 0: Number of entries in the object.
Sub-Index 1: COB-ID used for the PDO. Bit 31 set disables the PDO.
Sub-Index 2: Transmission Type – sets to Synchronous if set to Value 1-240. Sets to asynchronous for values 244-255.
Sub-Index 3: Inhibit time in units of 100 microseconds.
Sub-Index 4: Reserved and set to 00h
Sub-Index 5: Event time in milliseconds for RPDOs (used for Transmission Type 254 and 255).

5.16 Object 1401h: RPDO2 Communication Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1401h	0	Number of Entries	05h	No	Unsigned 8-bit	RO	Yes
1401h	1	COB-ID	00000301h	No	Unsigned 32-bit	RW	Yes
1401h	2	Transmission Type	FFh	No	Unsigned 8-bit	RW	Yes
1401h	3	Inhibit Time	0000h	No	Unsigned 16-bit	RW	Yes
1401h	4	Reserved	00h	No	Unsigned 8-bit	RW	Yes
1401h	5	Event Timer	0000h	No	Unsigned 16-bit	RW	Yes

This object contains the communication parameters for RPDO 1.
Sub-Index 0: Number of entries in the object.
Sub-Index 1: COB-ID used for the PDO. Bit 31 set disables the PDO.
Sub-Index 2: Transmission Type – sets to Synchronous if set to Value 1-240. Sets to asynchronous for values 244-255.
Sub-Index 3: Inhibit time in units of 100 microseconds.
Sub-Index 4: Reserved and set to 00h
Sub-Index 5: Event time in milliseconds for RPDOs (used for Transmission Type 254 and 255).

5.17 Object 1402h: RPDO3 Communication Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1402h	0	Number of Entries	05h	No	Unsigned 8-bit	RO	Yes
1402h	1	COB-ID	00000401h	No	Unsigned 32-bit	RW	Yes
1402h	2	Transmission Type	FFh	No	Unsigned 8-bit	RW	Yes
1402h	3	Inhibit Time	0000h	No	Unsigned 16-bit	RW	Yes
1402h	4	Reserved	00h	No	Unsigned 8-bit	RW	Yes
1402h	5	Event Timer	0000h	No	Unsigned 16-bit	RW	Yes

This object contains the communication parameters for RPDO3.
Sub-Index 0: Number of entries in the object.
Sub-Index 1: COB-ID used for the PDO. Bit 31 set disables the PDO.
Sub-Index 2: Transmission Type – sets to Synchronous if set to Value 1-240. Sets to asynchronous for values 244-255.
Sub-Index 3: Inhibit time in units of 100 microseconds.
Sub-Index 4: Reserved and set to 00h
Sub-Index 5: Event time in milliseconds for RPDOs (used for Transmission Type 254 and 255).

5.18 Object 1403h: RPDO4 Communication Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1403h	0	Number of Entries	05h	No	Unsigned 8-bit	RO	Yes
1403h	1	COB-ID	00000501h	No	Unsigned 32-bit	RW	Yes
1403h	2	Transmission Type	FFh	No	Unsigned 8-bit	RW	Yes
1403h	3	Inhibit Time	0000h	No	Unsigned 16-bit	RW	Yes
1403h	4	Reserved	00h	No	Unsigned 8-bit	RW	Yes
1403h	5	Event Timer	0000h	No	Unsigned 16-bit	RW	Yes

This object contains the communication parameters for RPDO4.
Sub-Index 0: Number of entries in the object.
Sub-Index 1: COB-ID used for the PDO. Bit 31 set disables the PDO.
Sub-Index 2: Transmission Type – sets to Synchronous if set to Value 1-240. Sets to asynchronous for values 244-255.
Sub-Index 3: Inhibit time in units of 100 microseconds.
Sub-Index 4: Reserved and set to 00h
Sub-Index 5: Event time in milliseconds for RPDOs (used for Transmission Type 254 and 255).

5.19 Object 1404h: RPDO5 Communication Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1404h	0	Number of Entries	05h	No	Unsigned 8-bit	RO	Yes
1404h	1	COB-ID	80000000h	No	Unsigned 32-bit	RW	Yes
1404h	2	Transmission Type	FFh	No	Unsigned 8-bit	RW	Yes
1404h	3	Inhibit Time	0000h	No	Unsigned 16-bit	RW	Yes
1404h	4	Reserved	00h	No	Unsigned 8-bit	RW	Yes
1404h	5	Event Timer	0000h	No	Unsigned 16-bit	RW	Yes

This object contains the communication parameters for RPDO5.
Sub-Index 0: Number of entries in the object.
Sub-Index 1: COB-ID used for the PDO. Bit 31 set disables the PDO.
Sub-Index 2: Transmission Type – sets to Synchronous if set to Value 1-240. Sets to asynchronous for values 244-255.
Sub-Index 3: Inhibit time in units of 100 microseconds.
Sub-Index 4: Reserved and set to 00h
Sub-Index 5: Event time in milliseconds for RPDOs (used for Transmission Type 254 and 255).

5.20 Object 1405h: RPDO6 Communication Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1405h	0	Number of Entries	05h	No	Unsigned 8-bit	RO	Yes
1405h	1	COB-ID	80000000h	No	Unsigned 32-bit	RW	Yes
1405h	2	Transmission Type	FFh	No	Unsigned 8-bit	RW	Yes
1405h	3	Inhibit Time	0000h	No	Unsigned 16-bit	RW	Yes
1405h	4	Reserved	00h	No	Unsigned 8-bit	RW	Yes
1405h	5	Event Timer	0000h	No	Unsigned 16-bit	RW	Yes

This object contains the communication parameters for RPDO6.
Sub-Index 0: Number of entries in the object.
Sub-Index 1: COB-ID used for the PDO. Bit 31 set disables the PDO.
Sub-Index 2: Transmission Type – sets to Synchronous if set to Value 1-240. Sets to asynchronous for values 244-255.
Sub-Index 3: Inhibit time in units of 100 microseconds.
Sub-Index 4: Reserved and set to 00h
Sub-Index 5: Event time in milliseconds for RPDOs (used for Transmission Type 254 and 255).

5.21 Object 1406h: RPDO7 Communication Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1406h	0	Number of Entries	05h	No	Unsigned 8-bit	RO	Yes
1406h	1	COB-ID	80000000h	No	Unsigned 32-bit	RW	Yes
1406h	2	Transmission Type	FFh	No	Unsigned 8-bit	RW	Yes
1406h	3	Inhibit Time	0000h	No	Unsigned 16-bit	RW	Yes
1406h	4	Reserved	00h	No	Unsigned 8-bit	RW	Yes
1406h	5	Event Timer	0000h	No	Unsigned 16-bit	RW	Yes

This object contains the communication parameters for RPDO7.
Sub-Index 0: Number of entries in the object.
Sub-Index 1: COB-ID used for the PDO. Bit 31 set disables the PDO.
Sub-Index 2: Transmission Type – sets to Synchronous if set to Value 1-240. Sets to asynchronous for values 244-255.
Sub-Index 3: Inhibit time in units of 100 microseconds.
Sub-Index 4: Reserved and set to 00h
Sub-Index 5: Event time in milliseconds for RPDOs (used for Transmission Type 254 and 255).

5.22 Object 1407h: RPDO8 Communication Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1407h	0	Number of Entries	05h	No	Unsigned 8-bit	RO	Yes
1407h	1	COB-ID	80000000h	No	Unsigned 32-bit	RW	Yes
1407h	2	Transmission Type	FFh	No	Unsigned 8-bit	RW	Yes
1407h	3	Inhibit Time	0h	No	Unsigned 16-bit	RW	Yes
1407h	4	Reserved	00h	No	Unsigned 8-bit	RW	Yes
1407h	5	Event Timer	0000h	No	Unsigned 16-bit	RW	Yes

This object contains the communication parameters for RPDO8.
Sub-Index 0: Number of entries in the object.
Sub-Index 1: COB-ID used for the PDO. Bit 31 set disables the PDO.
Sub-Index 2: Transmission Type – sets to Synchronous if set to Value 1-240. Sets to asynchronous for values 244-255.
Sub-Index 3: Inhibit time in units of 100 microseconds.
Sub-Index 4: Reserved and set to 00h
Sub-Index 5: Event time in milliseconds for RPDOs (used for Transmission Type 254 and 255).

5.23 Object 1600h: RPDO1 Mapping Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1600h	0	Number of Entries	03h	No	Unsigned 8-bit	RO	Yes
1600h	1	Mapping Entry 1	60FF0020h	No	Unsigned 32-bit	RW	Yes
1600h	2	Mapping Entry 2	60400010h	No	Unsigned 32-bit	RW	Yes
1600h	3	Mapping Entry 3	20000010h	No	Unsigned 32-bit	RW	Yes
1600h	4	Mapping Entry 4	00000000h	No	Unsigned 32-bit	RW	Yes
1600h	5	Mapping Entry 5	00000000h	No	Unsigned 32-bit	RW	Yes
1600h	6	Mapping Entry 6	00000000h	No	Unsigned 32-bit	RW	Yes
1600h	7	Mapping Entry 7	00000000h	No	Unsigned 32-bit	RW	Yes
1600h	8	Mapping Entry 8	00000000h	No	Unsigned 32-bit	RW	Yes

This object controls the objects mapped to RPDO1.
The bit fields of Mapping Entries 1-8 are defined below:

Data Field			
D4	D5	D6	D7
Object Index		Sub-Index	Length

5.24 Object 1601h: RPDO2 Mapping Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1601h	0	Number of Entries	02h	No	Unsigned 8-bit	RO	Yes
1601h	1	Mapping Entry 1	60C10120h	No	Unsigned 32-bit	RW	Yes
1601h	2	Mapping Entry 2	60400010h	No	Unsigned 32-bit	RW	Yes
1601h	3	Mapping Entry 3	00000000h	No	Unsigned 32-bit	RW	Yes
1601h	4	Mapping Entry 4	00000000h	No	Unsigned 32-bit	RW	Yes
1601h	5	Mapping Entry 5	00000000h	No	Unsigned 32-bit	RW	Yes
1601h	6	Mapping Entry 6	00000000h	No	Unsigned 32-bit	RW	Yes
1601h	7	Mapping Entry 7	00000000h	No	Unsigned 32-bit	RW	Yes
1601h	8	Mapping Entry 8	00000000h	No	Unsigned 32-bit	RW	Yes

This object controls the objects mapped to RPDO2.
The bit fields of Mapping Entries 1-8 are defined below:

Data Field			
D4	D5	D6	D7
Object Index		Sub-Index	Length

5.25 Object 1602h: RPDO3 Mapping Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1602h	0	Number of Entries	02h	No	Unsigned 8-bit	RO	Yes
1602h	1	Mapping Entry 1	607A0020h	No	Unsigned 32-bit	RW	Yes
1602h	2	Mapping Entry 2	60400010h	No	Unsigned 32-bit	RW	Yes
1602h	3	Mapping Entry 3	00000000h	No	Unsigned 32-bit	RW	Yes
1602h	4	Mapping Entry 4	00000000h	No	Unsigned 32-bit	RW	Yes
1602h	5	Mapping Entry 5	00000000h	No	Unsigned 32-bit	RW	Yes
1602h	6	Mapping Entry 6	00000000h	No	Unsigned 32-bit	RW	Yes
1602h	7	Mapping Entry 7	00000000h	No	Unsigned 32-bit	RW	Yes
1602h	8	Mapping Entry 8	00000000h	No	Unsigned 32-bit	RW	Yes

This object controls the objects mapped to RPDO3.
The bit fields of Mapping Entries 1-8 are defined below:

Data Field			
D4	D5	D6	D7
Object Index		Sub-Index	Length

5.26 Object 1603h: RPDO4 Mapping Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1603h	0	Number of Entries	02h	No	Unsigned 8-bit	RO	Yes
1603h	1	Mapping Entry 1	60400010h	No	Unsigned 32-bit	RW	Yes
1603h	2	Mapping Entry 2	60710010h	No	Unsigned 32-bit	RW	Yes
1603h	3	Mapping Entry 3	00000000h	No	Unsigned 32-bit	RW	Yes
1603h	4	Mapping Entry 4	00000000h	No	Unsigned 32-bit	RW	Yes
1603h	5	Mapping Entry 5	00000000h	No	Unsigned 32-bit	RW	Yes
1603h	6	Mapping Entry 6	00000000h	No	Unsigned 32-bit	RW	Yes
1603h	7	Mapping Entry 7	00000000h	No	Unsigned 32-bit	RW	Yes
1603h	8	Mapping Entry 8	00000000h	No	Unsigned 32-bit	RW	Yes

This object controls the objects mapped to RPDO4.
The bit fields of Mapping Entries 1-8 are defined below:

Data Field			
D4	D5	D6	D7
Object Index		Sub-Index	Length

5.27 Object 1604h: RPDO5 Mapping Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1604h	0	Number of Entries	01h	No	Unsigned 8-bit	RO	Yes
1604h	1	Mapping Entry 1	20000010h	No	Unsigned 32-bit	RW	Yes
1604h	2	Mapping Entry 2	00000000h	No	Unsigned 32-bit	RW	Yes
1604h	3	Mapping Entry 3	00000000h	No	Unsigned 32-bit	RW	Yes
1604h	4	Mapping Entry 4	00000000h	No	Unsigned 32-bit	RW	Yes
1604h	5	Mapping Entry 5	00000000h	No	Unsigned 32-bit	RW	Yes
1604h	6	Mapping Entry 6	00000000h	No	Unsigned 32-bit	RW	Yes
1604h	7	Mapping Entry 7	00000000h	No	Unsigned 32-bit	RW	Yes
1604h	8	Mapping Entry 8	00000000h	No	Unsigned 32-bit	RW	Yes

This object controls the objects mapped to RPDO5.
The bit fields of Mapping Entries 1-8 are defined below:

Data Field			
D4	D5	D6	D7
Object Index		Sub-Index	Length

5.28 Object 1605h: RPDO6 Mapping Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1605h	0	Number of Entries	01h	No	Unsigned 8-bit	RO	Yes
1605h	1	Mapping Entry 1	20000010h	No	Unsigned 32-bit	RW	Yes
1605h	2	Mapping Entry 2	00000000h	No	Unsigned 32-bit	RW	Yes
1605h	3	Mapping Entry 3	00000000h	No	Unsigned 32-bit	RW	Yes
1605h	4	Mapping Entry 4	00000000h	No	Unsigned 32-bit	RW	Yes
1605h	5	Mapping Entry 5	00000000h	No	Unsigned 32-bit	RW	Yes
1605h	6	Mapping Entry 6	00000000h	No	Unsigned 32-bit	RW	Yes
1605h	7	Mapping Entry 7	00000000h	No	Unsigned 32-bit	RW	Yes
1605h	8	Mapping Entry 8	00000000h	No	Unsigned 32-bit	RW	Yes

This object controls the objects mapped to RPDO6.
The bit fields of Mapping Entries 1-8 are defined below:

Data Field			
D4	D5	D6	D7
Object Index		Sub-Index	Length

5.29 Object 1606h: RPDO7 Mapping Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1606h	0	Number of Entries	01h	No	Unsigned 8-bit	RO	Yes
1606h	1	Mapping Entry 1	20000010h	No	Unsigned 32-bit	RW	Yes
1606h	2	Mapping Entry 2	00000000h	No	Unsigned 32-bit	RW	Yes
1606h	3	Mapping Entry 3	00000000h	No	Unsigned 32-bit	RW	Yes
1606h	4	Mapping Entry 4	00000000h	No	Unsigned 32-bit	RW	Yes
1606h	5	Mapping Entry 5	00000000h	No	Unsigned 32-bit	RW	Yes
1606h	6	Mapping Entry 6	00000000h	No	Unsigned 32-bit	RW	Yes
1606h	7	Mapping Entry 7	00000000h	No	Unsigned 32-bit	RW	Yes
1606h	8	Mapping Entry 8	00000000h	No	Unsigned 32-bit	RW	Yes

This object controls the objects mapped to RPDO7.
The bit fields of Mapping Entries 1-8 are defined below:

Data Field			
D4	D5	D6	D7
Object Index		Sub-Index	Length

5.30 Object 1607h: RPDO8 Mapping Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1607	0	Number of Entries	01h	No	Unsigned 8-bit	RO	Yes
1607	1	Mapping Entry 1	20000010h	No	Unsigned 32-bit	RW	Yes
1607	2	Mapping Entry 2	00000000h	No	Unsigned 32-bit	RW	Yes
1607	3	Mapping Entry 3	00000000h	No	Unsigned 32-bit	RW	Yes
1607	4	Mapping Entry 4	00000000h	No	Unsigned 32-bit	RW	Yes
1607	5	Mapping Entry 5	00000000h	No	Unsigned 32-bit	RW	Yes
1607	6	Mapping Entry 6	00000000h	No	Unsigned 32-bit	RW	Yes
1607	7	Mapping Entry 7	00000000h	No	Unsigned 32-bit	RW	Yes
1607	8	Mapping Entry 8	00000000h	No	Unsigned 32-bit	RW	Yes

This object controls the objects mapped to RPDO8.
The bit fields of Mapping Entries 1-8 are defined below:

Data Field			
D4	D5	D6	D7
Object Index		Sub-Index	Length

5.31 Object 1800h: TPDO1 Communication Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1800h	0	Number of Entries	05h	No	Unsigned 8-bit	RO	Yes
1800h	1	COB-ID	40000181h	No	Unsigned 32-bit	RW	Yes
1800h	2	Transmission Type	FFh	No	Unsigned 8-bit	RW	Yes
1800h	3	Inhibit Time	0000h	No	Unsigned 16-bit	RW	Yes
1800h	4	Reserved	00h	No	Unsigned 8-bit	RW	Yes
1800h	5	Event Timer	0000h	No	Unsigned 16-bit	RW	Yes

This object contains the communication parameters for TPDO1.
Sub-Index 0: Number of entries in the object.
Sub-Index 1: COB-ID used for the PDO. Bit 31 disables the PDO. No RTR allowed – Bit 30 set to 1.
Sub-Index 2: Transmission Type
Sub-Index 3: Inhibit time in units of 100 microseconds.
Sub-Index 4: Reserved and set to 00h.
Sub-Index 5: Event time in milliseconds for TPDOs (used for Transmission Type 254 and 255).

5.32 Object 1801h: TPDO2 Communication Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1801h	0	Number of Entries	05h	No	Unsigned 8-bit	RO	Yes
1801h	1	COB-ID	40000281h	No	Unsigned 32-bit	RW	Yes
1801h	2	Transmission Type	FFh	No	Unsigned 8-bit	RW	Yes
1801h	3	Inhibit Time	0000h	No	Unsigned 16-bit	RW	Yes
1801h	4	Reserved	00h	No	Unsigned 8-bit	RW	Yes
1801h	5	Event Timer	0000h	No	Unsigned 16-bit	RW	Yes

This object contains the communication parameters for TPDO2.
Sub-Index 0: Number of entries in the object.
Sub-Index 1: COB-ID used for the PDO. Bit 31 disables the PDO. No RTR allowed – Bit 30 set to 1.
Sub-Index 2: Transmission Type
Sub-Index 3: Inhibit time in units of 100 microseconds.
Sub-Index 4: Reserved and set to 00h.
Sub-Index 5: Event time in milliseconds for TPDOs (used for Transmission Type 254 and 255).

5.33 Object 1802h: TPDO3 Communication Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1802h	0	Number of Entries	05h	No	Unsigned 8-bit	RO	Yes
1802h	1	COB-ID	40000381h	No	Unsigned 32-bit	RW	Yes
1802h	2	Transmission Type	FFh	No	Unsigned 8-bit	RW	Yes
1802h	3	Inhibit Time	0000h	No	Unsigned 16-bit	RW	Yes
1802h	4	Reserved	00h	No	Unsigned 8-bit	RW	Yes
1802h	5	Event Timer	0000h	No	Unsigned 16-bit	RW	Yes

This object contains the communication parameters for TPDO3.

Sub-Index 0: Number of entries in the object.

Sub-Index 1: COB-ID used for the PDO. Bit 31 disables the PDO. No RTR allowed – Bit 30 set to 1.

Sub-Index 2: Transmission Type

Sub-Index 3: Inhibit time in units of 100 microseconds.

Sub-Index 4: Reserved and set to 00h.

Sub-Index 5: Event time in milliseconds for TPDOs (used for Transmission Type 254 and 255).

5.34 Object 1803h: TPDO4 Communication Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1803h	0	Number of Entries	05h	No	Unsigned 8-bit	RO	Yes
1803h	1	COB-ID	40000481h	No	Unsigned 32-bit	RW	Yes
1803h	2	Transmission Type	FFh	No	Unsigned 8-bit	RW	Yes
1803h	3	Inhibit Time	0000h	No	Unsigned 16-bit	RW	Yes
1803h	4	Reserved	00h	No	Unsigned 8-bit	RW	Yes
1803h	5	Event Timer	0000h	No	Unsigned 16-bit	RW	Yes

This object contains the communication parameters for TPDO4.
Sub-Index 0: Number of entries in the object.
Sub-Index 1: COB-ID used for the PDO. Bit 31 disables the PDO. No RTR allowed – Bit 30 set to 1.
Sub-Index 2: Transmission Type
Sub-Index 3: Inhibit time in units of 100 microseconds.
Sub-Index 4: Reserved and set to 00h.
Sub-Index 5: Event time in milliseconds for TPDOs (used for Transmission Type 254 and 255).

5.35 Object 1804h: TPDO5 Communication Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1804h	0	Number of Entries	05h	No	Unsigned 8-bit	RO	Yes
1804h	1	COB-ID	C0000000h	No	Unsigned 32-bit	RW	Yes
1804h	2	Transmission Type	00h	No	Unsigned 8-bit	RW	Yes
1804h	3	Inhibit Time	0000h	No	Unsigned 16-bit	RW	Yes
1804h	4	Reserved	00h	No	Unsigned 8-bit	RW	Yes
1804h	5	Event Timer	0000h	No	Unsigned 16-bit	RW	Yes

This object contains the communication parameters for TPDO5.
Sub-Index 0: Number of entries in the object.
Sub-Index 1: COB-ID used for the PDO. Bit 31 disables the PDO. No RTR allowed – Bit 30 set to 1.
Sub-Index 2: Transmission Type
Sub-Index 3: Inhibit time in units of 100 microseconds.
Sub-Index 4: Reserved and set to 00h.
Sub-Index 5: Event time in milliseconds for TPDOs (used for Transmission Type 254 and 255).

5.36 Object 1805h: TPDO6 Communication Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1805h	0	Number of Entries	05h	No	Unsigned 8-bit	RO	Yes
1805h	1	COB-ID	C0000000h	No	Unsigned 32-bit	RW	Yes
1805h	2	Transmission Type	00h	No	Unsigned 8-bit	RW	Yes
1805h	3	Inhibit Time	0000h	No	Unsigned 16-bit	RW	Yes
1805h	4	Compatibility Entry	00h	No	Unsigned 8-bit	RW	Yes
1805h	5	Event Timer	0000h	No	Unsigned 16-bit	RW	Yes

This object contains the communication parameters for TPDO6.
Sub-Index 0: Number of entries in the object.
Sub-Index 1: COB-ID used for the PDO. Bit 31 disables the PDO. No RTR allowed – Bit 30 set to 1.
Sub-Index 2: Transmission Type
Sub-Index 3: Inhibit time in units of 100 microseconds.
Sub-Index 4: Reserved and set to 00h.
Sub-Index 5: Event time in milliseconds for TPDOs (used for Transmission Type 254 and 255).

5.37 Object 1806h: TPDO7 Communication Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1806h	0	Number of Entries	05h	No	Unsigned 8-bit	RO	Yes
1806h	1	COB-ID	C0000000h	No	Unsigned 32-bit	RW	Yes
1806h	2	Transmission Type	00h	No	Unsigned 8-bit	RW	Yes
1806h	3	Inhibit Time	0000h	No	Unsigned 16-bit	RW	Yes
1806h	4	Reserved	00h	No	Unsigned 8-bit	RW	Yes
1806h	5	Event Timer	0000h	No	Unsigned 16-bit	RW	Yes

This object contains the communication parameters for TPDO7.
Sub-Index 0: Number of entries in the object.
Sub-Index 1: COB-ID used for the PDO. Bit 31 disables the PDO. No RTR allowed – Bit 30 set to 1.
Sub-Index 2: Transmission Type
Sub-Index 3: Inhibit time in units of 100 microseconds.
Sub-Index 4: Reserved and set to 00h.
Sub-Index 5: Event time in milliseconds for TPDOs (used for Transmission Type 254 and 255).

5.38 Object 1807h: TPDO8 Communication Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1807h	0	Number of Entries	05h	No	Unsigned 8-bit	RO	Yes
1807h	1	COB-ID	C0000000h	No	Unsigned 32-bit	RW	Yes
1807h	2	Transmission Type	00h	No	Unsigned 8-bit	RW	Yes
1807h	3	Inhibit Time	0000h	No	Unsigned 16-bit	RW	Yes
1807h	4	Reserved	00h	No	Unsigned 8-bit	RW	Yes
1807h	5	Event Timer	0000h	No	Unsigned 16-bit	RW	Yes

This object contains the communication parameters for TPDO8.
Sub-Index 0: Number of entries in the object.
Sub-Index 1: COB-ID used for the PDO. Bit 31 set disables the PDO.
Sub-Index 2: Transmission Type
Sub-Index 3: Inhibit time in units of 100 microseconds.
Sub-Index 4: Reserved and set to 00h.
Sub-Index 5: Event time in milliseconds for TPDOs (used for Transmission Type 254 and 255).

5.39 Object 1A00h: TPDO1 Mapping Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1A00	0	Number of Entries	03h	No	Unsigned 8-bit	RO	Yes
1A00	1	Mapping Entry 1	60410010h	No	Unsigned 32-bit	RW	Yes
1A00	2	Mapping Entry 2	60640020h	No	Unsigned 32-bit	RW	Yes
1A00	3	Mapping Entry 3	20010010h	No	Unsigned 32-bit	RW	Yes
1A00	4	Mapping Entry 4	00000000h	No	Unsigned 32-bit	RW	Yes
1A00	5	Mapping Entry 5	00000000h	No	Unsigned 32-bit	RW	Yes
1A00	6	Mapping Entry 6	00000000h	No	Unsigned 32-bit	RW	Yes
1A00	7	Mapping Entry 7	00000000h	No	Unsigned 32-bit	RW	Yes
1A00	8	Mapping Entry 8	00000000h	No	Unsigned 32-bit	RW	Yes

This object controls the objects mapped to TPDO1.
The bit fields of Mapping Entries 1-8 are defined below:

Data Field			
D4	D5	D6	D7
Object Index		Sub-Index	Length

5.40 Object 1A01h: TPDO2 Mapping Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1A01	0	Number of Entries	01h	No	Unsigned 8-bit	RO	Yes
1A01	1	Mapping Entry 1	20010010h	No	Unsigned 32-bit	RW	Yes
1A01	2	Mapping Entry 2	00000000h	No	Unsigned 32-bit	RW	Yes
1A01	3	Mapping Entry 3	00000000h	No	Unsigned 32-bit	RW	Yes
1A01	4	Mapping Entry 4	00000000h	No	Unsigned 32-bit	RW	Yes
1A01	5	Mapping Entry 5	00000000h	No	Unsigned 32-bit	RW	Yes
1A01	6	Mapping Entry 6	00000000h	No	Unsigned 32-bit	RW	Yes
1A01	7	Mapping Entry 7	00000000h	No	Unsigned 32-bit	RW	Yes
1A01	8	Mapping Entry 8	00000000h	No	Unsigned 32-bit	RW	Yes

This object controls the objects mapped to TPDO2.
The bit fields of Mapping Entries 1-8 are defined below:

Data Field			
D4	D5	D6	D7
Object Index		Sub-Index	Length

5.41 Object 1A02h: TPDO3 Mapping Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1A02	0	Number of Entries	01h	No	Unsigned 8-bit	RO	Yes
1A02	1	Mapping Entry 1	60640020h	No	Unsigned 32-bit	RW	Yes
1A02	2	Mapping Entry 2	00000000h	No	Unsigned 32-bit	RW	Yes
1A02	3	Mapping Entry 3	00000000h	No	Unsigned 32-bit	RW	Yes
1A02	4	Mapping Entry 4	00000000h	No	Unsigned 32-bit	RW	Yes
1A02	5	Mapping Entry 5	00000000h	No	Unsigned 32-bit	RW	Yes
1A02	6	Mapping Entry 6	00000000h	No	Unsigned 32-bit	RW	Yes
1A02	7	Mapping Entry 7	00000000h	No	Unsigned 32-bit	RW	Yes
1A02	8	Mapping Entry 8	00000000h	No	Unsigned 32-bit	RW	Yes

This object controls the objects mapped to TPDO3.
The bit fields of Mapping Entries 1-8 are defined below:

Data Field			
D4	D5	D6	D7
Object Index		Sub-Index	Length

5.42 Object 1A03h: TPDO4 Mapping Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1A03	0	Number of Entries	01h	No	Unsigned 8-bit	RO	Yes
1A03	1	Mapping Entry 1	60770010h	No	Unsigned 32-bit	RW	Yes
1A03	2	Mapping Entry 2	00000000h	No	Unsigned 32-bit	RW	Yes
1A03	3	Mapping Entry 3	00000000h	No	Unsigned 32-bit	RW	Yes
1A03	4	Mapping Entry 4	00000000h	No	Unsigned 32-bit	RW	Yes
1A03	5	Mapping Entry 5	00000000h	No	Unsigned 32-bit	RW	Yes
1A03	6	Mapping Entry 6	00000000h	No	Unsigned 32-bit	RW	Yes
1A03	7	Mapping Entry 7	00000000h	No	Unsigned 32-bit	RW	Yes
1A03	8	Mapping Entry 8	00000000h	No	Unsigned 32-bit	RW	Yes

This object controls the objects mapped to TPDO4.
The bit fields of Mapping Entries 1-8 are defined below:

Data Field			
D4	D5	D6	D7
Object Index		Sub-Index	Length

5.43 Object 1A04h: TPDO5 Mapping Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1A04	0	Number of Entries	01h	No	Unsigned 8-bit	RO	Yes
1A04	1	Mapping Entry 1	20010010h	No	Unsigned 32-bit	RW	Yes
1A04	2	Mapping Entry 2	00000000h	No	Unsigned 32-bit	RW	Yes
1A04	3	Mapping Entry 3	00000000h	No	Unsigned 32-bit	RW	Yes
1A04	4	Mapping Entry 4	00000000h	No	Unsigned 32-bit	RW	Yes
1A04	5	Mapping Entry 5	00000000h	No	Unsigned 32-bit	RW	Yes
1A04	6	Mapping Entry 6	00000000h	No	Unsigned 32-bit	RW	Yes
1A04	7	Mapping Entry 7	00000000h	No	Unsigned 32-bit	RW	Yes
1A04	8	Mapping Entry 8	00000000h	No	Unsigned 32-bit	RW	Yes

This object controls the objects mapped to TPDO5.
The bit fields of Mapping Entries 1-8 are defined below:

Data Field			
D4	D5	D6	D7
Object Index		Sub-Index	Length

5.44 Object 1A05h: TPDO6 Mapping Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1A05	0	Number of Entries	01h	No	Unsigned 8-bit	RO	Yes
1A05	1	Mapping Entry 1	20010010h	No	Unsigned 32-bit	RW	Yes
1A05	2	Mapping Entry 2	00000000h	No	Unsigned 32-bit	RW	Yes
1A05	3	Mapping Entry 3	00000000h	No	Unsigned 32-bit	RW	Yes
1A05	4	Mapping Entry 4	00000000h	No	Unsigned 32-bit	RW	Yes
1A05	5	Mapping Entry 5	00000000h	No	Unsigned 32-bit	RW	Yes
1A05	6	Mapping Entry 6	00000000h	No	Unsigned 32-bit	RW	Yes
1A05	7	Mapping Entry 7	00000000h	No	Unsigned 32-bit	RW	Yes
1A05	8	Mapping Entry 8	00000000h	No	Unsigned 32-bit	RW	Yes

This object controls the objects mapped to TPDO6.
The bit fields of Mapping Entries 1-8 are defined below:

Data Field			
D4	D5	D6	D7
Object Index		Sub-Index	Length

5.45 Object 1A06h: TPDO7 Mapping Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1A06	0	Number of Entries	01h	No	Unsigned 8-bit	RO	Yes
1A06	1	Mapping Entry 1	20010010h	No	Unsigned 32-bit	RW	Yes
1A06	2	Mapping Entry 2	00000000h	No	Unsigned 32-bit	RW	Yes
1A06	3	Mapping Entry 3	00000000h	No	Unsigned 32-bit	RW	Yes
1A06	4	Mapping Entry 4	00000000h	No	Unsigned 32-bit	RW	Yes
1A06	5	Mapping Entry 5	00000000h	No	Unsigned 32-bit	RW	Yes
1A06	6	Mapping Entry 6	00000000h	No	Unsigned 32-bit	RW	Yes
1A06	7	Mapping Entry 7	00000000h	No	Unsigned 32-bit	RW	Yes
1A06	8	Mapping Entry 8	00000000h	No	Unsigned 32-bit	RW	Yes

This object controls the objects mapped to TPDO7.
The bit fields of Mapping Entries 1-8 are defined below:

Data Field			
D4	D5	D6	D7
Object Index		Sub-Index	Length

5.46 Object 1A07h: TPDO8 Mapping Parameter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1A07	0	Number of Entries	01h	No	Unsigned 8-bit	RO	Yes
1A07	1	Mapping Entry 1	20010010h	No	Unsigned 32-bit	RW	Yes
1A07	2	Mapping Entry 2	00000000h	No	Unsigned 32-bit	RW	Yes
1A07	3	Mapping Entry 3	00000000h	No	Unsigned 32-bit	RW	Yes
1A07	4	Mapping Entry 4	00000000h	No	Unsigned 32-bit	RW	Yes
1A07	5	Mapping Entry 5	00000000h	No	Unsigned 32-bit	RW	Yes
1A07	6	Mapping Entry 6	00000000h	No	Unsigned 32-bit	RW	Yes
1A07	7	Mapping Entry 7	00000000h	No	Unsigned 32-bit	RW	Yes
1A07	8	Mapping Entry 8	00000000h	No	Unsigned 32-bit	RW	Yes

This object controls the objects mapped to TPDO8.
The bit fields of Mapping Entries 1-8 are defined below:

Data Field			
D4	D5	D6	D7
Object Index		Sub-Index	Length

5.47 Object 1F80h: NMT Startup

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
1F80h	0	NMT Startup	0h	No	Unsigned 32-bit	RW	No

This object dictates the state after starting the drive. ‘0h’ is for starting in operational.

6. Connecting to a Glentek CANopen Drive

6.1 RJ45 Connection

The CANopen interface can be used via the HOST, COM1, or COM2 connector. Glentek Gamma Series Drives employ the CANopen protocol that is based on the CAN Physical Layer as described in the CAN in Automation (CiA) standard DS-301 V4.02. This port is the primary means of communication with the CANopen network for real-time control. The Gamma drives can maintain high speed (up to 1 Mb/s) serial communication interface for communications between nodes in real-time control applications. The drive device profile is based on IEC61800-7-201/301 (CiA 402).

The port utilizes an RJ45 type connector, Illustrated in Figure 6-1. Table 6-1 shows the RJ45 Pin designations and Figure 6-2 shows the CANopen Connection Diagram.

Table 6-1: RJ45 CANopen Communication Designations

Pin #	Pin Type	Pin Name	Pin Function (Isolated Version)	Pin Function (Non-Isolated Version)
1	Input/output	CAN_H	Dominant High	Dominant High
2	Input/output	CAN_L	Dominant Low	Dominant Low
3	Common	CAN GND	Ground	Common (Digital)
4	Reserved	Reserved	Reserved	Reserved
5	Reserved	Reserved	Reserved	Reserved
6	Reserved	Reserved	Reserved	Reserved
7	Common	CAN GND	Ground	Common (Digital)
8	Reserved	Reserved	Reserved	Reserved

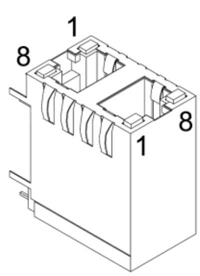


Figure 6-1: RJ45 Double Mating Connector

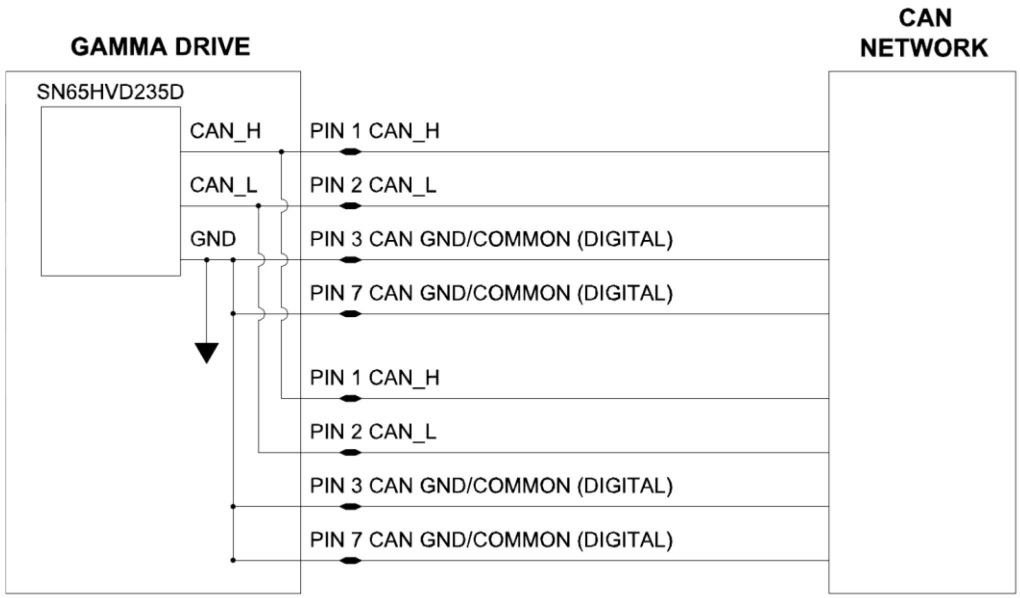


Figure 6-2: CANopen Connection Diagram

6.2 Baud Rate Select

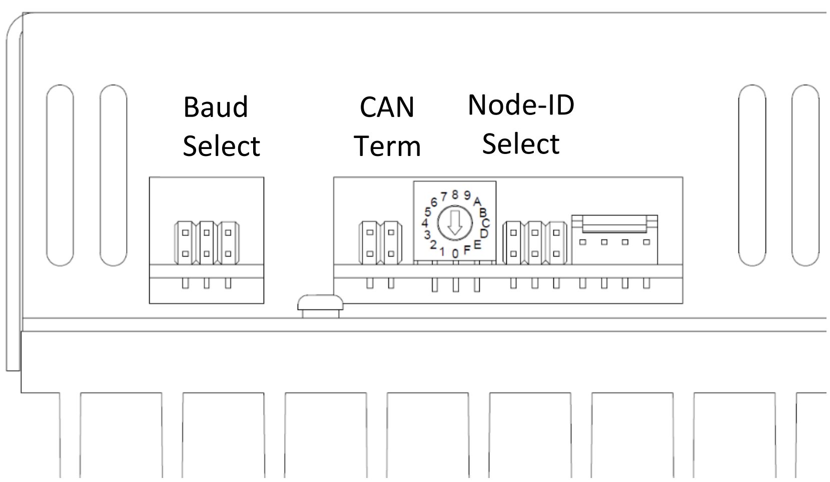


Figure 6-3: Side Panel of Gamma Drive

The Baud Rate, CAN Termination and Node-ID can be configured using the connectors in Figure 6-3: Side Panel of Gamma Drive.

Table 6-2: Serial Speed Over Distance

Baud Rate (Bits/second)	Max CAN bus cable length (meters)	Baud Rate Jumper Position
1 M	25 (82 ft)	0
500 K	100 (328 ft)	1
250 K	250 (820 ft)	2
125 K	500 (1640 ft)	3
100 K	-	4
50 K	1000 (3280 ft)	5

Due to impedance effects, the baud rate must be decreased as the length of the CAN Bus increases. The maximum suggested Baud Rate based on CAN bus length is given in Table 6-2: Serial Speed Over Distance. In addition, Table 6-2: Serial Speed Over Distance[1] shows the Jumper Position necessary to configure the baud rate.

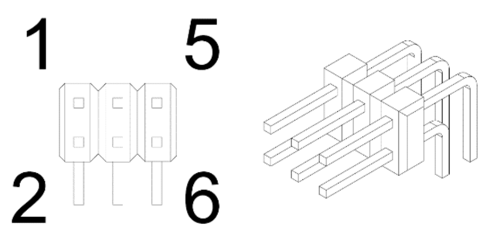


Figure 6-4: Baud Rate Selection

Jumper Position is defined as the binary equivalent of closing (or connecting) pins 1-2, 3-4, and 5-6. For example, a jumper connecting pins 1-2 and 3-4 corresponds to Jumper Position 3. Similarly, a jumper connecting pins 1-2 and 5-6 corresponds to Jumper Position 5.

Table 6-3: Selecting Jumper Position

Pins 1-2	Pins 3-4	Pins 5-6	Jumper Position
Open	Open	Open	0
Closed	Open	Open	1
Open	Closed	Open	2
Closed	Closed	Open	3
Open	Open	Closed	4
Closed	Open	Closed	5

6.3 CAN Termination

A 120 [Ω] termination resistor is required at the last drive node in the CANopen network to prevent reflection. This resistor is provided inside the drive for convenience. The drive is terminated by placing a two-pin micro-shunt jumper on pins 3 and 4 of the termination jumper connector.

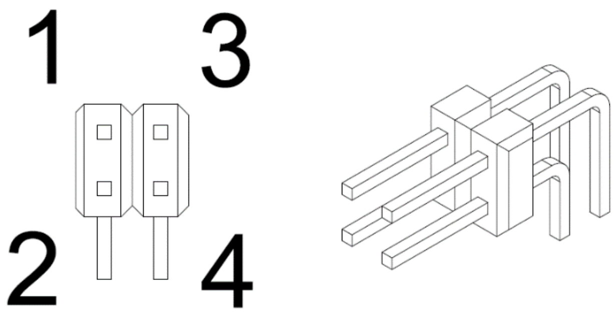


Figure 6-5: CAN Termination Jumper

6.4 Node-ID Select

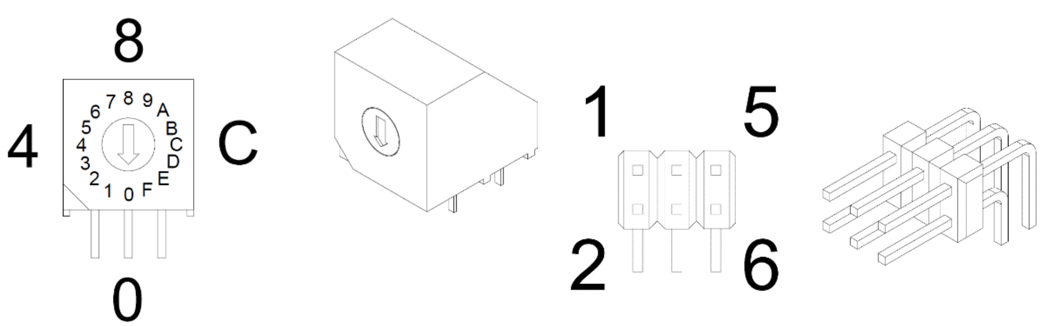


Figure 6-6: Node-ID Select Dial/Jumper

The valid range of CANopen addresses extend from 1-127. The Node-ID of the Gamma drive can be assigned via the 16-bit rotary switch and the address select jumpers located beside.

$$\text{Node} - \text{ID} = \mathit{Rotary\,Value} + 16 \times \mathit{Jumper\ Position}$$

The Jumper Position uses the same scheme outlined in Table 6-3: Selecting Jumper Position; however, the Node-ID Jumper should be used instead of the Baud Rate Jumper. For example, setting the Jumper Position = 2 and the Rotary Value = Eh will set the Node-ID to be 46. Special care must be taken so that the chosen Node-ID is unused by other slaves on the CAN bus and not equal to zero. After power cycling, the gamma drive will assume the configured Node-ID during the NMT boot up state.

Table 6-4: Node-ID Selection

Node-ID #	Address Select Rotary Switch	Node-ID Jumper Position
0-15	0h-Fh	0
16-31	0h-Fh	1
32-47	0h-Fh	2
48-63	0h-Fh	3
64-79	0h-Fh	4
80-95	0h-Fh	5
96-111	0h-Fh	6
112-127	0h-Fh	7

7. Drive Control and Objects

7.1 General

Figure 7-1 specifies how the Glentek Drive operates locally or via the network remotely. The drive is operated by local signals and by the controlword sent by the control device via the network. The state of the drive is reported by the statusword produced by the drive device. The drive State Machine is also controlled by error detection signals.

The Drive State Machine defines the status and the possible control sequence of the drive. A single state represents a special internal or external behavior. The state of the drive determines which commands are accepted. For example, it is only possible to start a point-to-point move when the drive is in the operation enabled state.

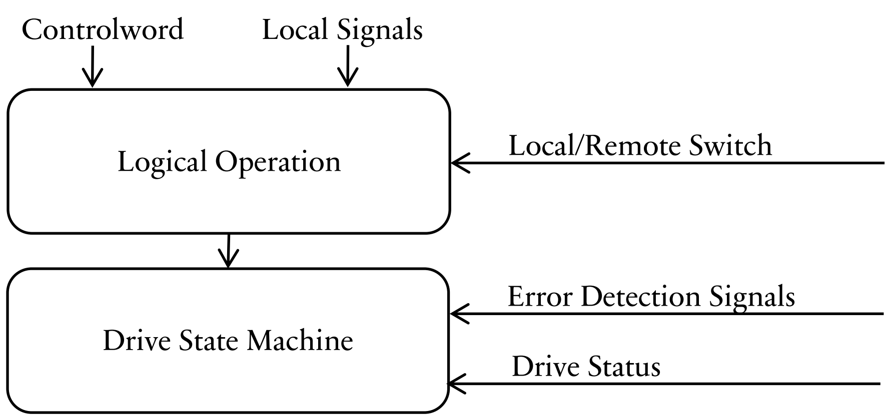


Figure 7-1: Remote and Local Control

7.2 Drive State Machine

Figure 7-2 shows the Drive State Machine with respect to control of the power electronics as a result of user commands and internal drive faults.

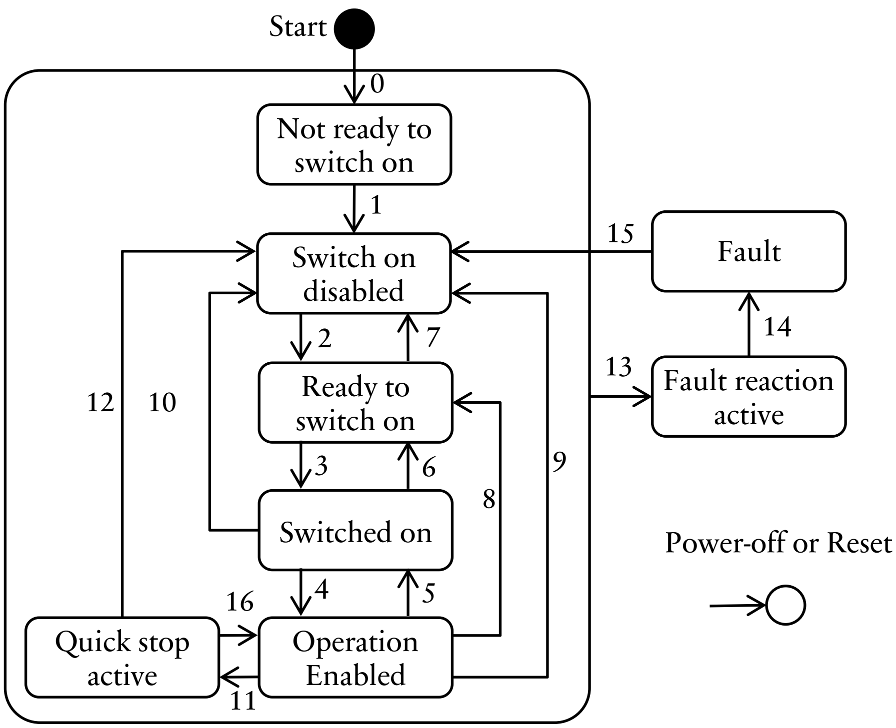


Figure 7-2: Drive State Machine

Table 7-1 describes the drive functions in each state. If in the *quick stop active* state, the quick stop option code is set to 5, 6, 7 or 8, the drive device will not leave the state but will transit to the *operation enabled* state with the *enabled operation* command. Object 6040h: Controlword controls State Machine state.

Table 7-1: State Machine State Functions

Drive State	Function Description
Not Ready to Switch on	Logic Power is applied to the Drive. Drive is initialized. Brake should be active if present. Drive is disabled. High Level Power may be applied.
Switch On Disabled	Drive Default parameters are loaded. Drive parameters may be changed. Drive is disabled. High Level Power may be applied. Drive has no fault. Brake should be active if present.
Ready to Switch On	Drive parameters may be changed. Drive is disabled. High Level Power may be applied. Drive has no fault. Brake should be active if present.
Switched On	Drive parameters may be changed. Drive is disabled. High Level Power is applied. Drive has no fault. Brake should be active if present. Drive is Ready for operation. No hardware Inhibit is active.
Operation Enabled	Limited Drive parameters may be changed depends on the operation mode. High Level Power is applied. Drive is enabled, and outputs power to the motor. Drive has no fault. Brake is released.
Quick Stop Active	Limited Drive parameters may be changed. High Level Power is applied. Drive is enabled, and outputs power to the motor. Drive has no fault. Brake is released. The transition State after Quick Stop completed depends on the 'Halt Option Code' 605Dh.
Fault Reaction Active	Faults have been detected. Drive reacts to the faults based on 'Fault reaction option code' 605Eh and fault type. Drive is disabled immediately when a Fatal fault is detected and the brake should be active if present.
Fault	Drive parameters may be changed. Drive is disabled. High Level Power may be applied depends on the application. Drive has faults. Brake should be active if present.

If a state transition is requested, the related actions is processed completely before transitioning to a new state. The drive provides option to control the contactor for the mains may switch the high-level power. If the high-level power is switched-off, the motor is free to rotate if not braked. If a brake is present, the high-level power is switched off after a delay time in order to allow the brake to engage. No energy is being supply to the motor when the drive is disabled. Drive function enabled implies that energy is being supplied to the motor. Target or set-point values, such as torque, velocity, position, are only processed while the drive function is enabled.

If a fault is detected in the drive device, there is a transition to the *fault reaction active* state. Drive is automatically disabled in Fault State. The *fault* state is only left by using a fault reset command, after the fault is removed. In case of a fatal error, the drive device is no longer able to control the motor, remove the high power immediately.

7.3 Control and Status Objects

7.3.1 Object 6040h: Controlword

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6040h	00h	Controlword	0	No	Unsigned 16-bit	RW	No

This object indicates the received command controlling the Drive State Machine. It is structed as defined in Figure 7-3 . The commands are coded as given in Table 7-2 .

15 … 11	10	9	8	7	6 … 4	3	2	1	0
Reserved	*r*	*oms*	*h*	*fr*	*oms*	*eo*	*qs*	*ev*	*so*
MSB									LSB

Reserved; **r** = reserved; **oms** = operation mode specific; **h** = halt; **fr** = fault reset; **eo** = enabled operation; **qs** = quick stop; **ev** = enable voltage; **so** = switch on

Figure 7-3: Controlword Value Definition

Table 7-2: Controlword Command Coding

Command	Bits of the Controlword					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	X	1	1	0	2, 6, 8
Switch on	0	0	1	1	1	3
Switch on + Enable operation	0	1	1	1	1	3 + 4*
Disable voltage	0	X	X	0	X	7, 9, 10, 12
Quick stop	0	X	0	1	X	7, 10, 11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset		X	X	X	X	15
*Automatic transition to Enable operation state after executing Switch on state functionality						

Bits 9, 6, 5, and 4 of the Controlword are operation mode specific. The halt function (bit 8) behavior is also operation mode specific. If bit 8 is set to 1, the commanded motion is interrupted, the drive behaves as defined in the halt option code. After releasing the halt function, the commanded motion will resume if possible. Bit 10 is reserved for further use and is set to 0. Bits 11-15 are reserved.

7.3.2 Object 6041h: Statusword

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6041h	00h	Statusword	0	No	Unsigned 16-bit	R	No

This object provides the status of the Drive. It is structed as defined in Figure 7-4. The commands are coded as given in Table 7-3.

15 14	13 12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	*oms*	*ila*	*tr*	*rm*	*ms*	*w*	*sod*	*qs*	*ve*	*f*	*oe*	*so*	*rtso*
MSB													LSB

Reserved; **oms** = operation mode specific; **ila** = Internal limit active; **tr** = target reached; **rm** = remote; **w** = warning; **sod** = switch on disable; **qs** = quick stop; **ve** = voltage enabled; **f** = fault; **oe** = operation enabled; **so** = switch on; **rtso** = ready to switch on

Figure 7-4: Statusword Value Definition

Table 7-3: Statusword Command Coding

Statusword	Drive State Machine state
xxxx xxxx x0xx 0000b	Not ready to switch on
xxxx xxxx x1xx 0000b	Switch on disabled
xxxx xxxx x01x 0001b	Ready to switch on
xxxx xxxx x01x 0011b	Switched on
xxxx xxxx x01x 0111b	Operation enabled
xxxx xxxx x00x 0111b	Quick stop active
xxxx xxxx x0xx 1111b	Fault reaction active
xxxx xxxx x0xx 1000b	Fault

If bit 4 (voltage enabled) of the statusword is 1 then high voltage is being applied to the drive. If bit 5 (quick stop) of the statusword is 0 then the drive is reacting on a quick stop request. If bit 7 (warning) is 1 then there is a warning condition present. A warning is not an error nor a fault. An example of a warning can be a temperature limit exceeded or a job being refused. The status of the Drive State Machine state does not change. The cause of the warning is given in Object 603Fh: Error Code.

If bit 9 (remote) is 1 then the controlword is processed. If it is 0 (local) then the controlword is not processed. The drive provides actual values and it may accept COB for configuration data transmission for other parameter objects. If bit 10 (target reached) is 1 then the drive has reached it is set point. The set point is operation mode specific. Bit 10 is always set to 1 when the operation mode is changed. The change of a target value by software alters this bit. If quick stop option code is 5, 6, 7, or 8, bit 10 is set to 1 when the quick stop operation is finished and the drive is halted.

If bit 11 (internal limit active) of the statusword is 1, then an internal limit is active. An example of an internal limit is the position range limit. Bit 13 and bit 12 of the statusword are operation mode specific. Bit 14 and bit 15 are reserved.

7.4 Operation Objects

The drive behavior depends on the active mode of operation. The control device writes to Object 6060h: Modes of Operation to indicate the actual activated operation mode. Controlword, Statusword and set-point objects are used mode-specific to avoid inconsistencies and erroneous behavior.

The following modes of operation are described in this manual:

- Homing mode
- Profile position mode (in progress)
- Interpolated position mode
- Profile velocity mode (e.g. servo drives)
- Profile Torque mode (in progress)
- Velocity mode (e.g. frequency converter) (in progress)
- Cyclic sync position mode
- Cyclic sync velocity mode
- Cyclic sync torque mode (in progress)

With the exception of ‘Homing mode’, the listed modes of operation deal with set-points.

7.4.1 Object 6060h: Modes of Operation

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6060h	00h	Modes of Operation	9	No	Integer 8-bit	RW	Yes

This object sets the operation mode. The object shows only the value of the requested operation mode. The actual operation mode of the drive is reflected in Object 6061h: Modes of Operation Display. Table 7-4 specifies the value definition.

Table 7-4: 6060h Value Range Definition

Value	Definition
-128to -1	Reserved
0	No mode change assigned
+1	Profile position mode (pp)
+2	Velocity mode (vl)
+3	Profile velocity mode (pv)
+4	Torque profile mode (tq)
+5	Reserved (r)
+6	Homing mode (hm)
+7	Interpolated position mode (ip)
+8	Cyclic sync position mode (csp)
+9	Cyclic sync velocity mode (csv)
+10	Cyclic sync torque mode (cst)
+11 to +127	Reserved

7.4.2 Object 6061h: Modes of Operation Display

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6061h	00h	Modes of Operation Display	9	No	Integer 8-bit	R	No

This object provides the actual operation mode. Table 7-4 specifies the value definition.

7.4.3 Object 6502h: Supported Drive Modes

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6502h	00h	Supported Drive Modes	1E4h	No	Unsigned 32-bit	R	No

This object provides the information on the supported drive modes. Figure 7-5 specifies the value definition.

31 … 10	9	8	7	6	5	4	3	2	1	0
Reserved	*cst*	*csv*	*csp*	*ip*	*hm*	*r*	*tq*	*pv*	*vl*	*pp*

MSB LSB

Legend: See Table 7-4

Figure 7-5: Object 6502h Value Definition

For bit 9 (cst), bit 8 (csv), bit 7 (csp), bit 6 (ip), bit 5 (hm), bit 3 (tq), bit 2 (pv), bit 1 (vl), and bit 0 (pp) a 1 means that the mode is supported and a 0 means that the mode is not supported. Bit 10 through bit 31 are set to 0, reserved.

7.4.4 Object 605Ah: Quick Stop Option Code

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
605Ah	00h	Quick Stop Option Code	2	No	Integer 16-bit	RW	Yes

This object indicates what action is performed when the quick stop function is executed. The slowdown ramp is the deceleration value of the used mode operations. Table 7-5 specifies the value definition.

Table 7-5: 605Ah Value Range Definition

Value	Definition
-32,786 to -1	Reserved
0	Disabled drive function
+1	Slow down on slow down ramp and transit into Switch on disabled
+2	Slow down on quick stop ramp and transit into Switch on disabled
+3	Slow down on current limit and transit into Switch on disabled
+4	Reserved
+5	Slow down on slow down ramp and stay in Quick stop active
+6	Slow down on quick stop ramp and stay in Quick stop active
+7	Slow down on current limit and stay in Quick stop active
+8	Reserved
+9 to +32,767	Reserved

7.4.5 Object 605Dh: Halt Option Code

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
605Dh	00h	Halt Option Code	1	No	Integer 16-bit	RW	Yes

This object indicates what action is performed when the halt function is executed. The slowdown ramp is the deceleration value of the used mode of operations Table 7-6 specifies the value definition.

Table 7-6: 605Dh Value Range Definition

Value	Definition
-32,768 to -1	Reserved
0	Reserved
+1	Slow down on slow down ramp and stay in Operation enabled
+2	Slow down on quick stop ramp and stay in Operation enabled
+3	Slow down on current limit and stay in Operation enabled
+4 to +32,767	Reserved

7.4.6 Object 603Fh: Error Code

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
603Fh	00h	Error Code	0	No	Unsigned 16-bit	R	No

This object provides the error code of the last error that occurred in the drive device. In the CANopen network, this object provides the same information as the lower 16-bit of sub-index 01h of the pre-defined Object 1003h: Error Field. See Table (List of Errors) for value range.

8. Homing Mode

8.1 Overview

For the operation of positioning drives, an exact knowledge of the absolute position is normally required. In homing mode, the drive seeks the home position (also known as datum, reference point, machine zero) to gain the exact knowledge of the absolute position. The different homing methods will be discussed in this chapter. Positive and negative limit switches at the end of travel, a home switch in mid-travel, and the index pulses from an incremental encoder are common feedbacks detected and utilized during homing.

8.2 Functional Description

Figure 8-1: Homing Mode Function shows the define input and output objects. The user may specify the speeds, acceleration and the method of homing. The object home offset allows the user to set the zero position away from the home position; see the definition of Object 607Ch: Home offset. There is no output data except for those bits in the statusword. The bits return the status, the result of the homing process, and the demand to the position control loops.

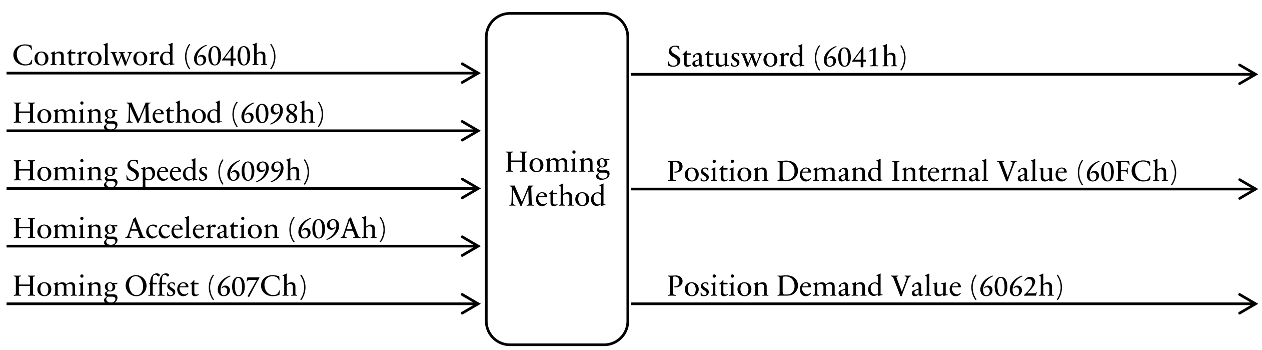


Figure 8-1: Homing Mode Function

There are four sources of homing signals available; the negative and positive limit switches, the home switch and the index pulse from an incremental encoder. Each method described in this section utilizes a unique combination of the homing signals to achieve the home attain status.

In the diagrams of the homing methods shown below, the encoder count increases as the axis' position moves to the right (positive direction) and the encoder count decreases as the axis' position moves to the left (negative direction). In other words, the left is the minimum position and the right is the maximum position. In case that a limit switch, at the end of travel, is activated by the axis, the axis will move in the opposite direction to leave the position.

There are two homing speeds in a typical homing cycle. The faster speed is used to find the home switch, which is denoted by thick black line in the diagrams, and the slower speed is used to find the index pulse, which is denoted by thin black line in the diagrams. The direction of movement is also indicated by an arrow-head. The red dot with the number indicates the homing attained position. Figure 8-2 highlights the key components utilized in the homing diagrams Figure 8-3 through Figure 8-32.

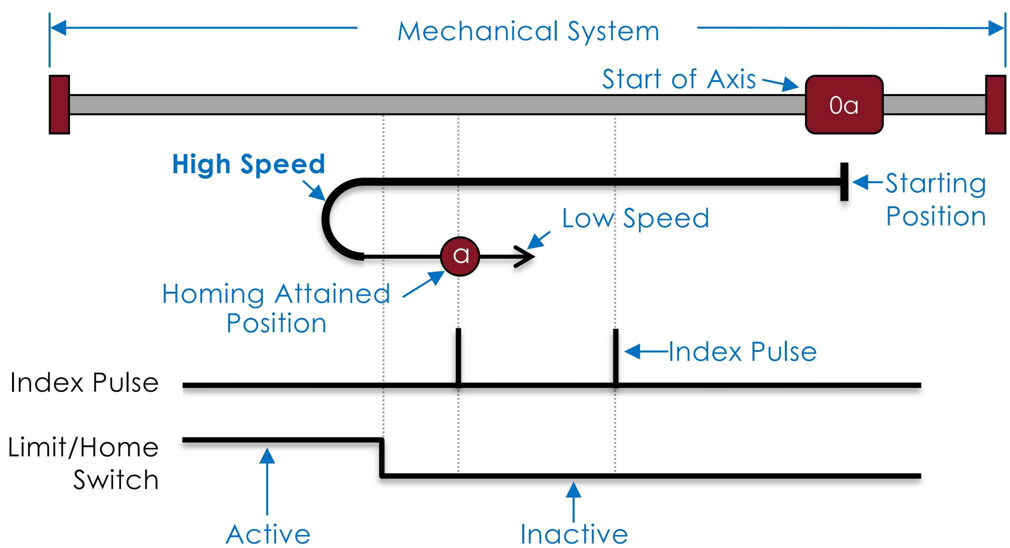


Figure 8-2: Key Components for the Homing Diagrams

8.3 General Definitions

This section describes the different homing methods.

- **Homing methods 1-14** utilize home and limit switches in combination with the index pulse.
- **Homing methods 17-30** have identical routines as those to methods 1-14 except that they do not utilize the index pulse and instead are dependent on the edge of the home or limit switch.
- **Homing methods 33 and 34** home to the nearest index pulse.
- **Homing method 35** homes to the current position

8.3.1 Method 1: Homing on Negative Limit Switch and Index Pulse

Axis 1a moves in the negative direction until the negative limit switch (NLS) is active. The axis will then move in the positive direction until the first index pulse is detected after the NLS goes inactive. After this event, the drive sets *home attained*.

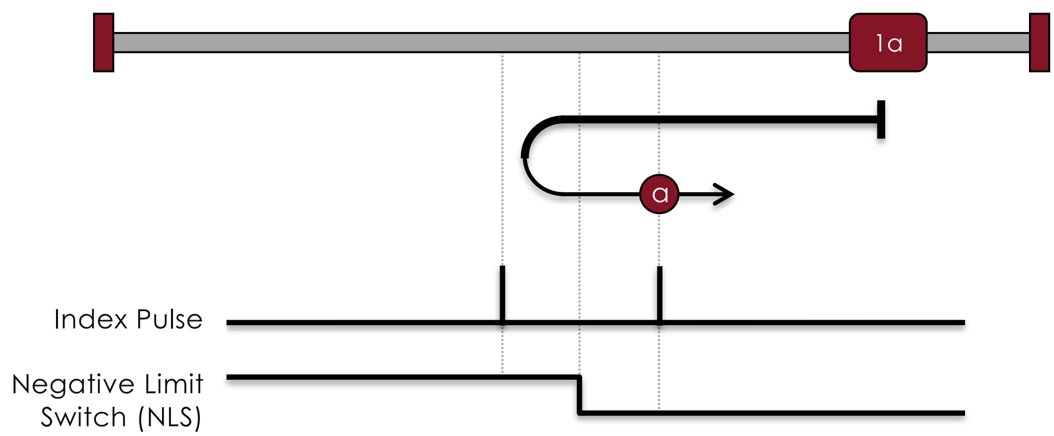


Figure 8-3: Homing Method 1

8.3.2 Method 2: Homing on Positive Limit Switch and Index Pulse

Axis 2a moves in the positive direction until the **positive limit switch** (PLS) is active. The axis will then move in the negative direction until the first index pulse is detected after the PLS goes inactive. After this event, the drive sets *home attained*.

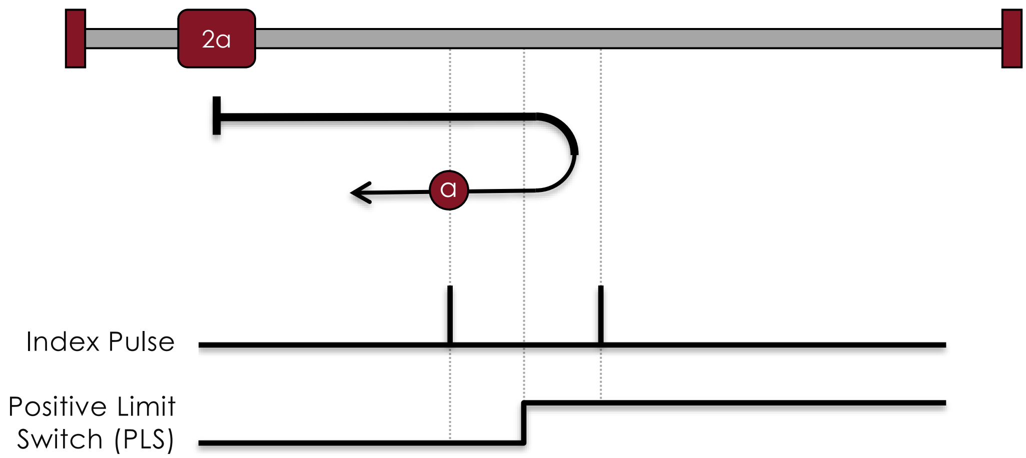


Figure 8-4: Homing Method 2

8.3.3 Method 3 and 4: Homing on Positive Home Switch and Index Pulse

Using these methods as shown in Figure 8-5 and Figure 8-6 the initial direction of movement is dependent on the state of the home switch. The home attain position should be at the index pulse to the left or right of the point where the home switch changes states. If the initial position is situated so that the direction of movement needs to be reversed during homing, the point at which it reverses should be after the home switch changes states.

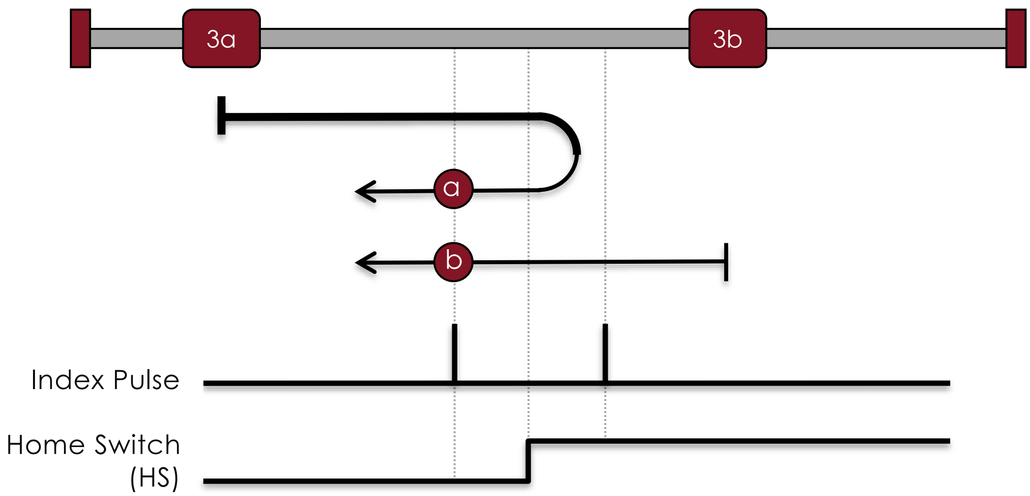


Figure 8-5: Homing Method 3

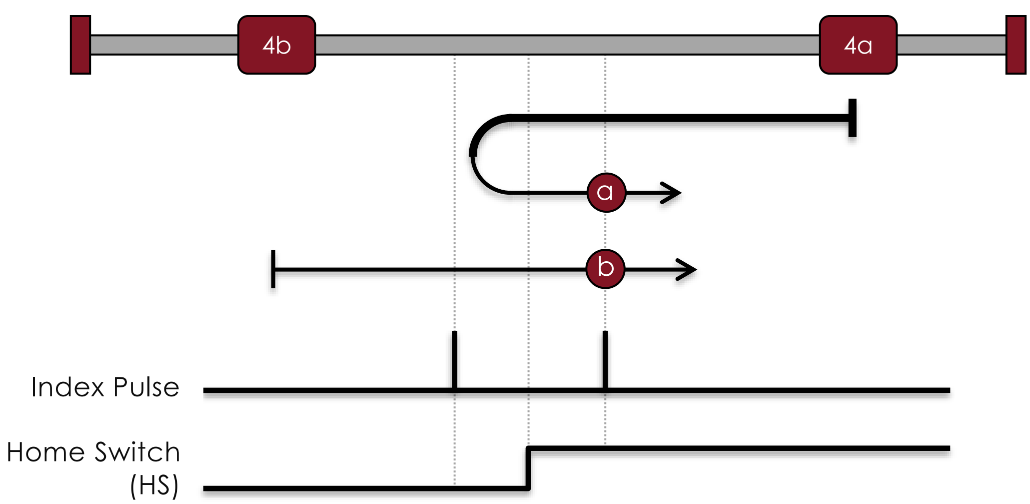


Figure 8-6: Homing Method 4

8.3.4 Method 5 and 6: Homing on Negative Home Switch and Index Pulse

Using these methods as shown in Figure 8-7 and Figure 8-8 the initial direction of movement is dependent on the state of the home switch. The home attain position should be at the index pulse to the left or right of the point where the home switch changes states. If the initial position is situated so that the direction of movement needs to be reversed during homing, the point at which it reverses should be after the home switch changes states.

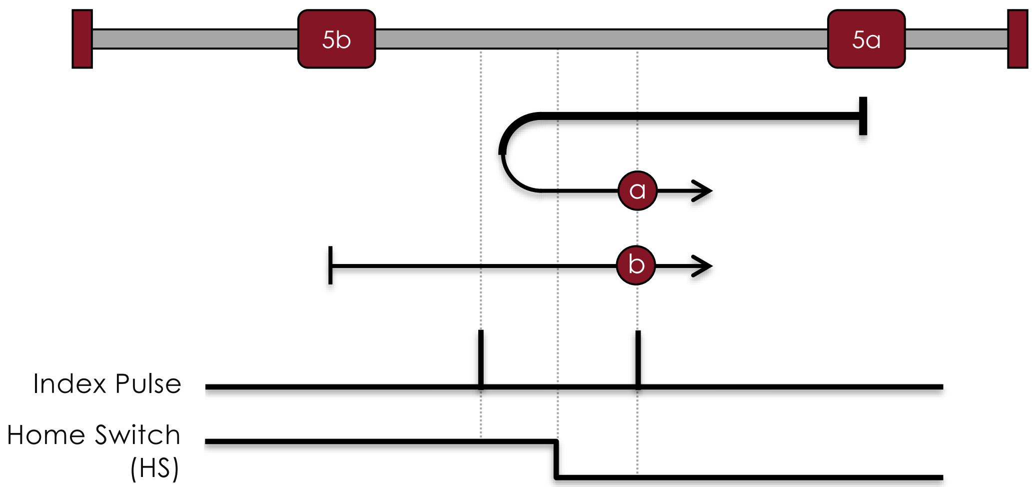


Figure 8-7: Homing Method 5

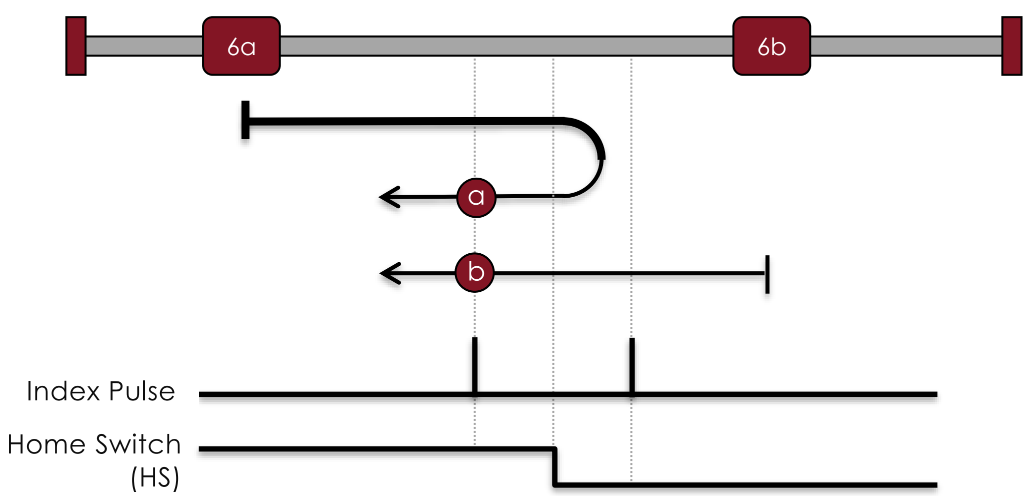


Figure 8-8: Homing Method 6

8.3.5 Method 7 to 14: Homing on Home Switch and Index Pulse

These methods use a home switch, which is active only a portion of the travel. The home switch has a 'momentary' action as the axis position goes pass it. Using methods 7 to 10, Figure 8-9 through Figure 8-12 respectively, the initial direction of movement is to the right. Using methods 11 to 14, Figure 8-13 through Figure 8-16 respectively, the initial direction of movement is to the left except if the home switch is active at the start of the motion. In such case, the initial direction of motion depends on the edge of interest. The home position is at the index pulse on the rising or falling edges of the home switch. If the initial direction of movement leads away from the home switch, the drive should reverse on encountering the relevant limit switch.

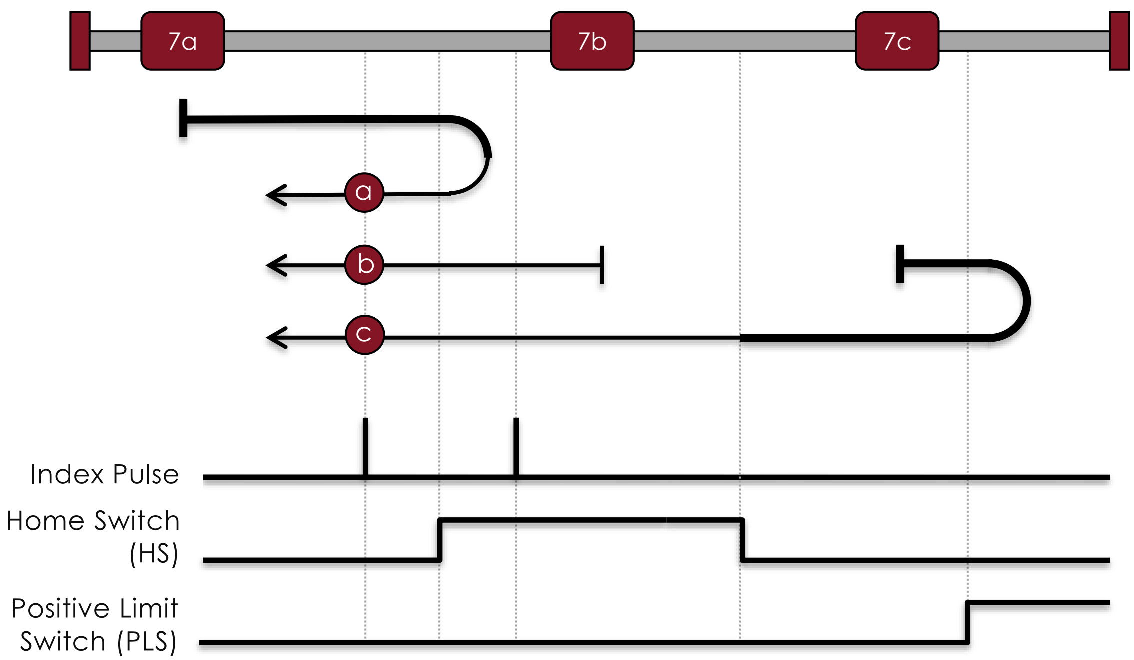


Figure 8-9: Homing Method 7

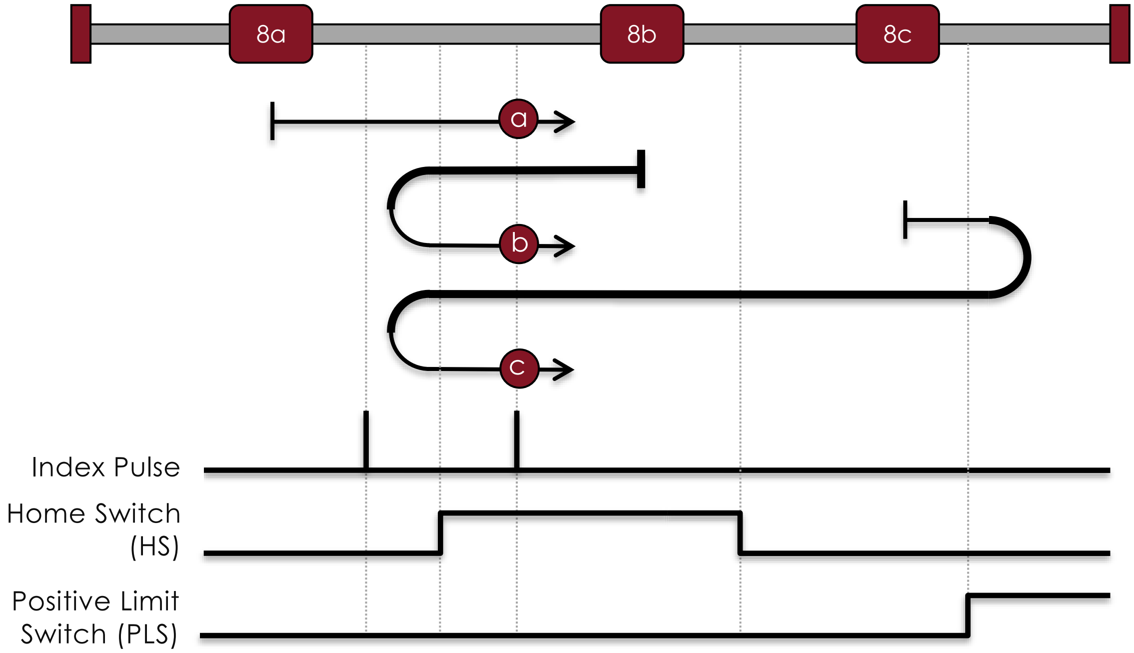


Figure 8-10: Homing Method 8

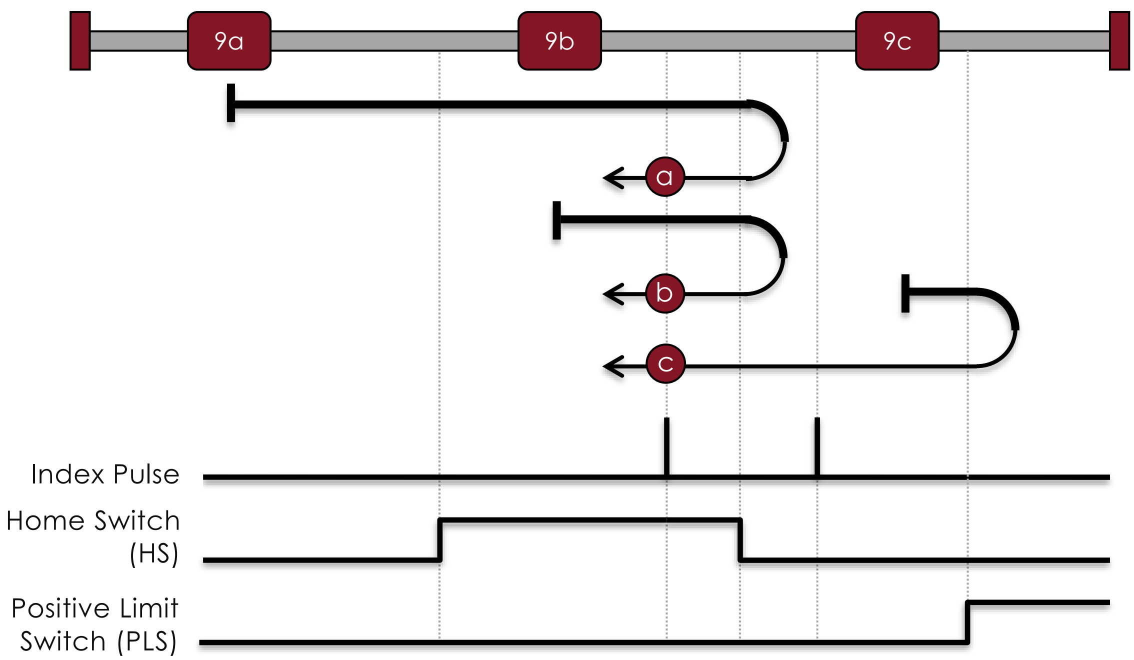


Figure 8-11: Homing Method 9

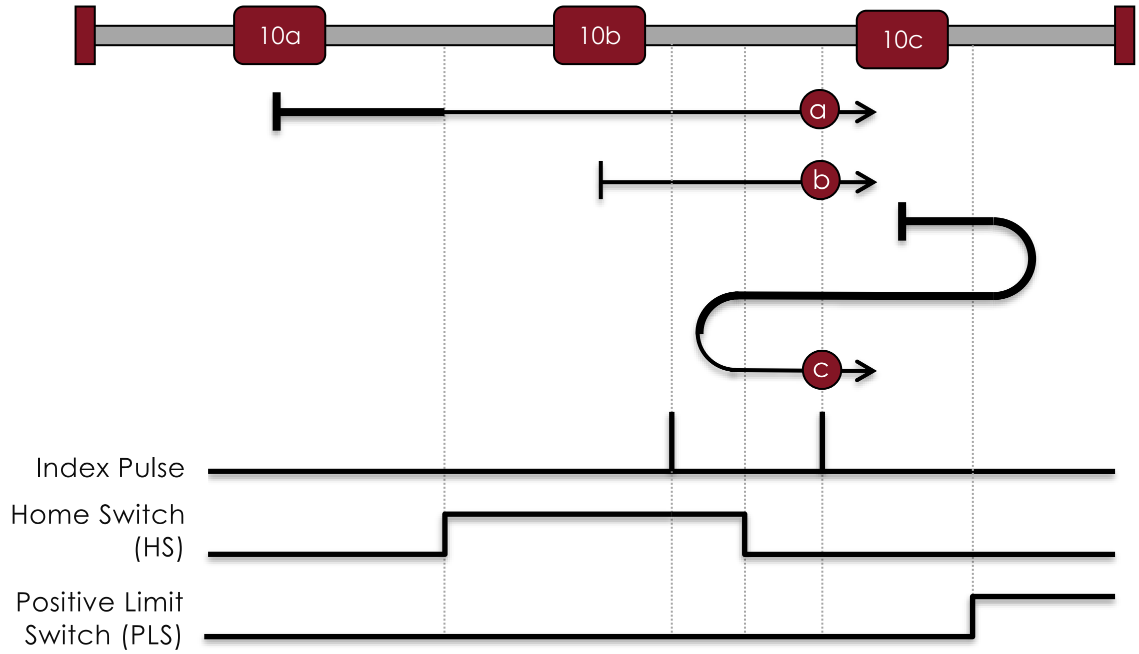


Figure 8-12: Homing Method 10

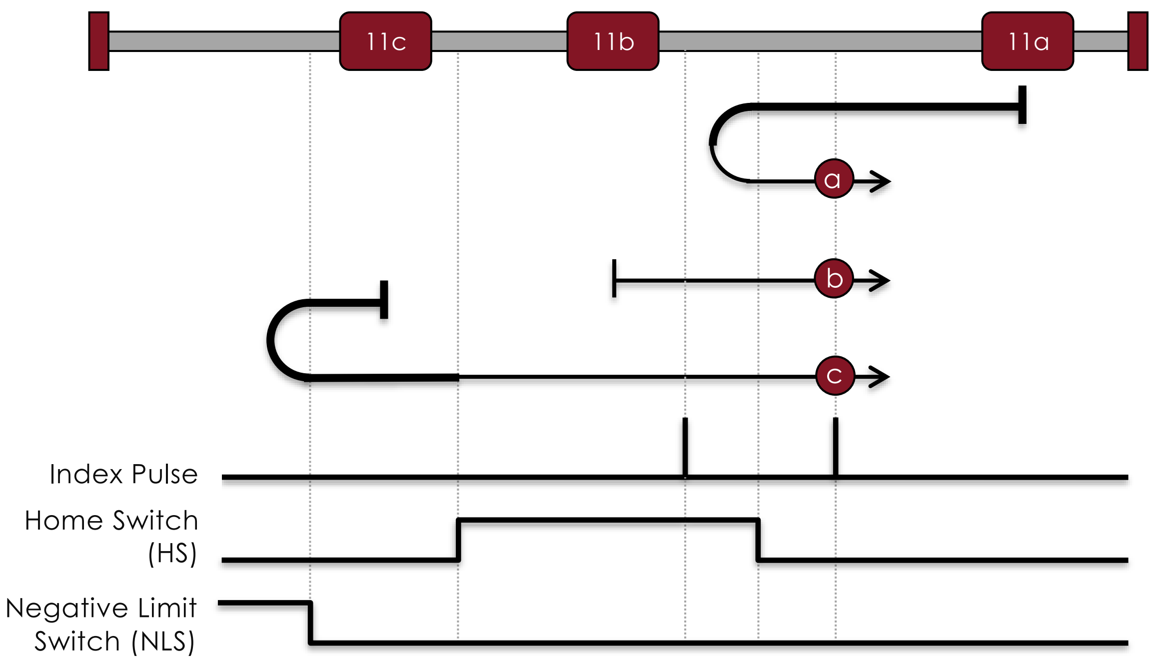


Figure 8-13: Homing Method 11

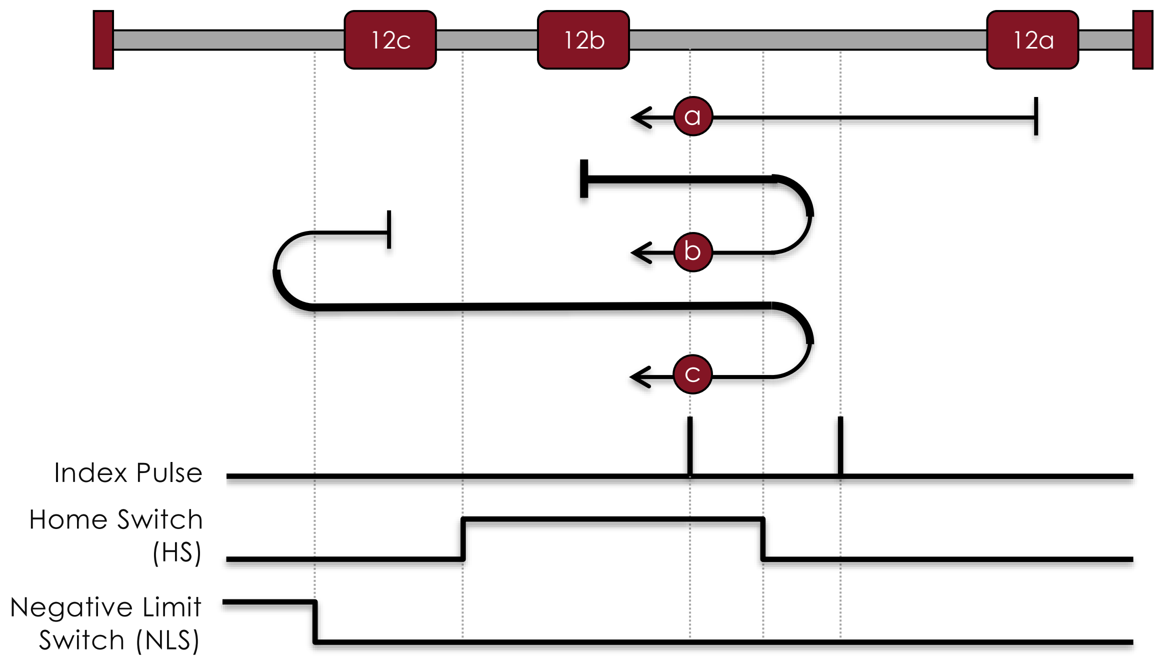


Figure 8-14: Homing Method 12

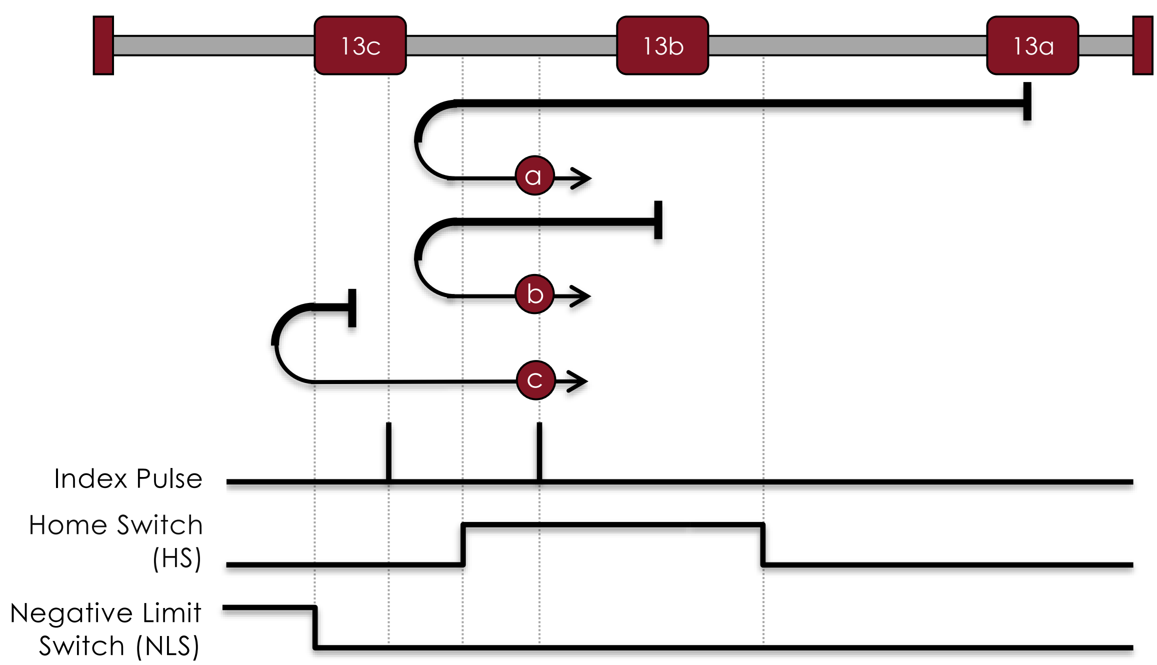


Figure 8-15: Homing Method 13

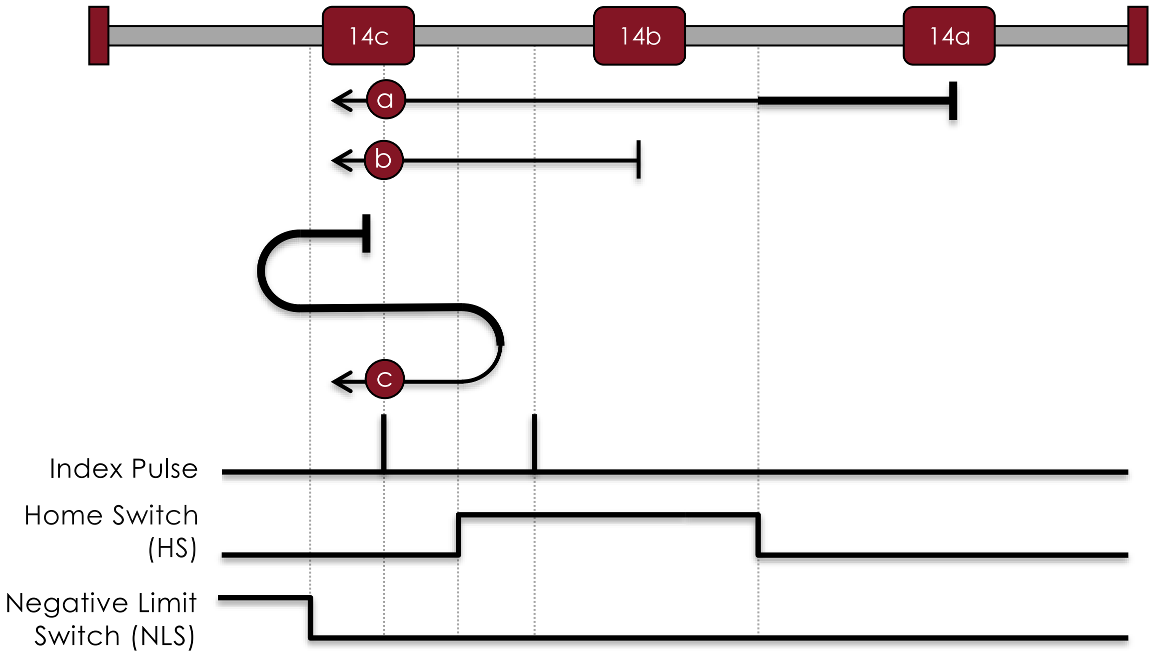


Figure 8-16: Homing Method 14

8.3.6 Method 15 and 16: Reserved

8.3.7 Method 17 to 30: Homing without Index Pulse

These methods are similar to methods 1 through 14 except that the home position is not dependent on the index pulse but only dependent on the relevant home or limit switch transitions. For example, methods 17 (Figure 8-17) and 18 (Figure 8-18) are similar to methods 1 (Figure 8-3) and 2 (Figure 8-4) respectively and so on.

Axis 17a moves in the negative direction until the negative limit switch (NLS) is active. The axis will then move in the positive direction until the NLS goes inactive. After this event, the drive sets *home attained*.

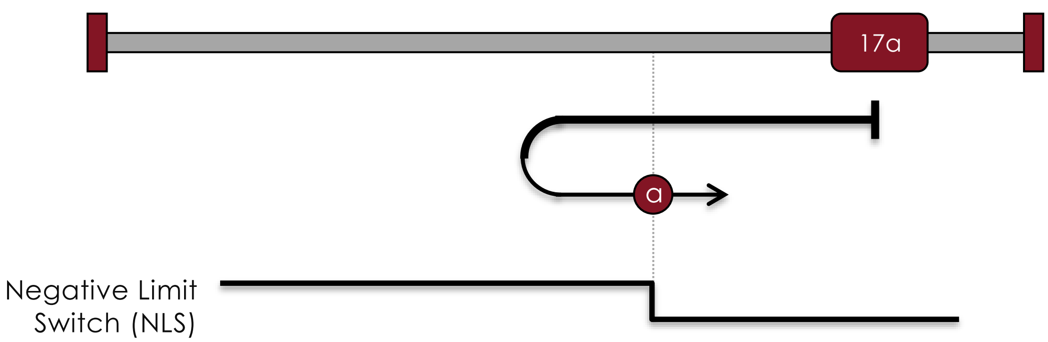


Figure 8-17: Homing Method 17

Axis 18a moves in the positive direction until the positive limit switch (PLS) is active. The axis will then move in the negative direction until the PLS goes inactive. After this event, the drive sets *home attained*.

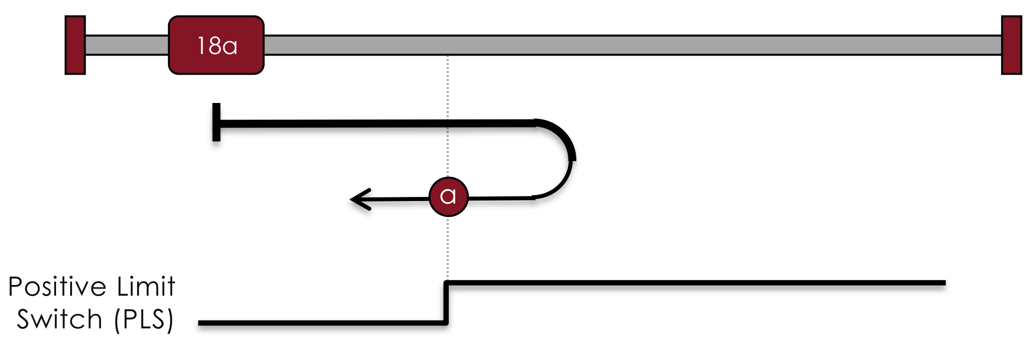


Figure 8-18: Homing Method 18

Using these methods as shown in Figure 8-19 and Figure 8-20 the initial direction of movement is dependent on the state of the home switch. The home attain position should be where the home switch changes states. If the initial position is situated so that the direction of movement needs to be reversed during homing, the point at which it reverses should be after the home switch changes states.

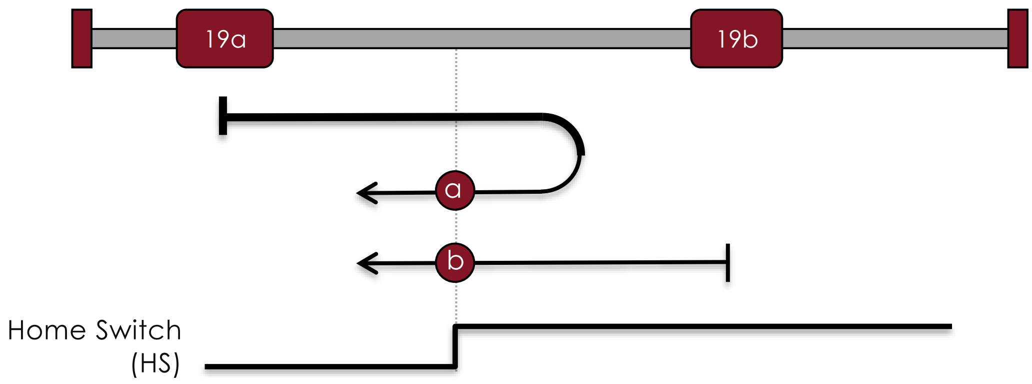


Figure 8-19: Homing Method 19

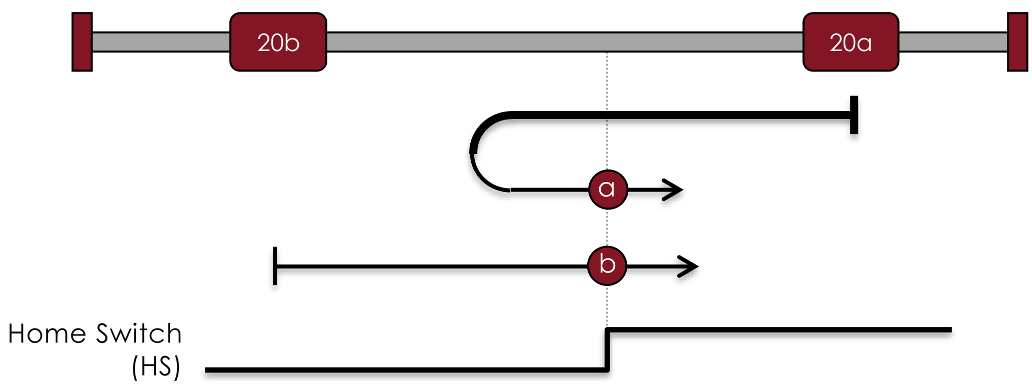


Figure 8-20: Homing Method 20

Using these methods as shown in Figure 8-21 and Figure 8-22 the initial direction of movement is dependent on the state of the home switch. The home attain position should be where the home switch changes states. If the initial position is situated so that the direction of movement needs to be reversed during homing, the point at which it reverses should be after the home switch changes states.

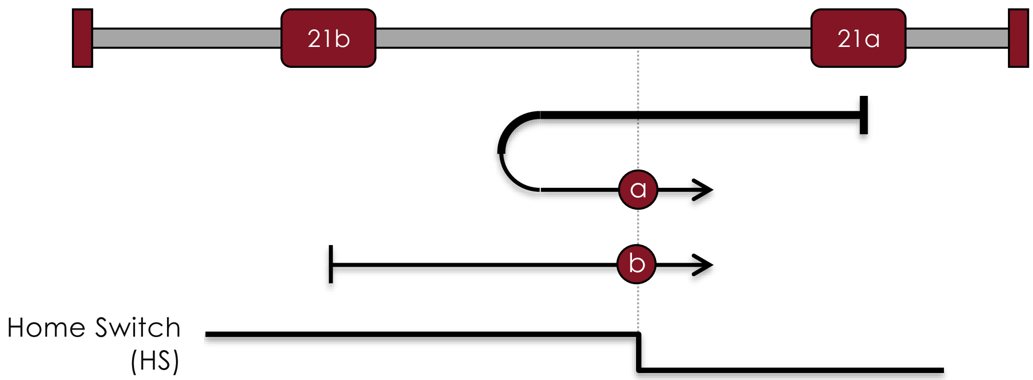


Figure 8-21: Homing Method 21

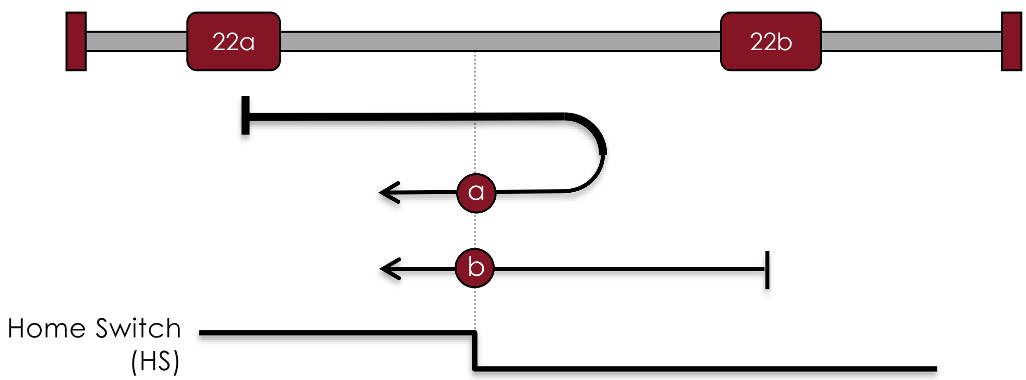


Figure 8-22: Homing Method 22

These methods use a home switch, which is active only a portion of the travel. The home switch has a 'momentary' action as the axis position goes pass it. Using methods 23 to 26, Figure 8-23 through Figure 8-26 respectively, the initial direction of movement is to the right. Using methods 27 to 30, Figure 8-27 through Figure 8-30 respectively, the initial direction of movement is to the left except if the home switch is active at the start of the motion. In such case, the initial direction of motion depends on the edge of interest. The home position is the rising or falling edges of the home switch. If the initial direction of movement leads away from the home switch, the drive should reverse on encountering the relevant limit switch.

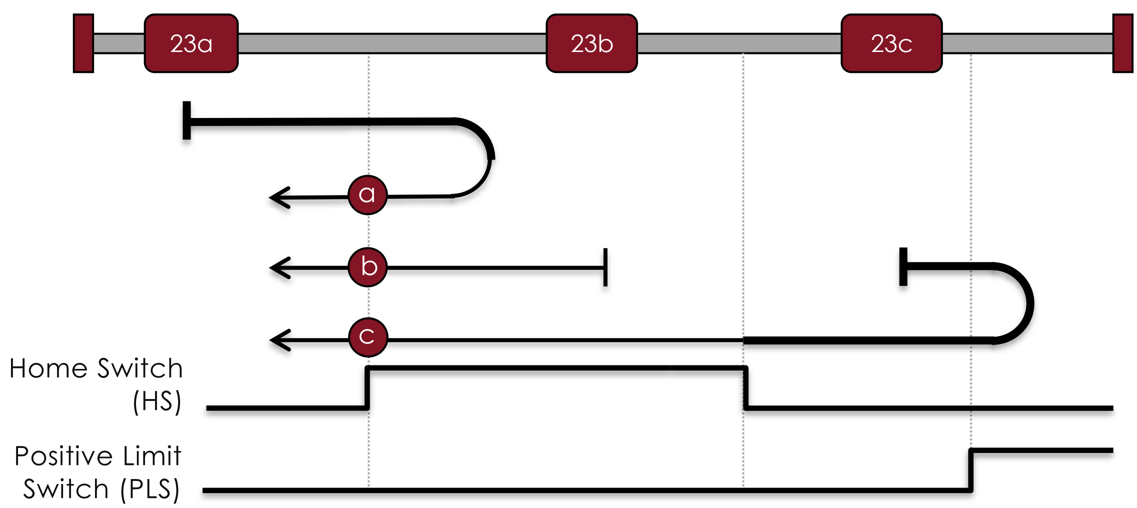


Figure 8-23: Homing Method 23

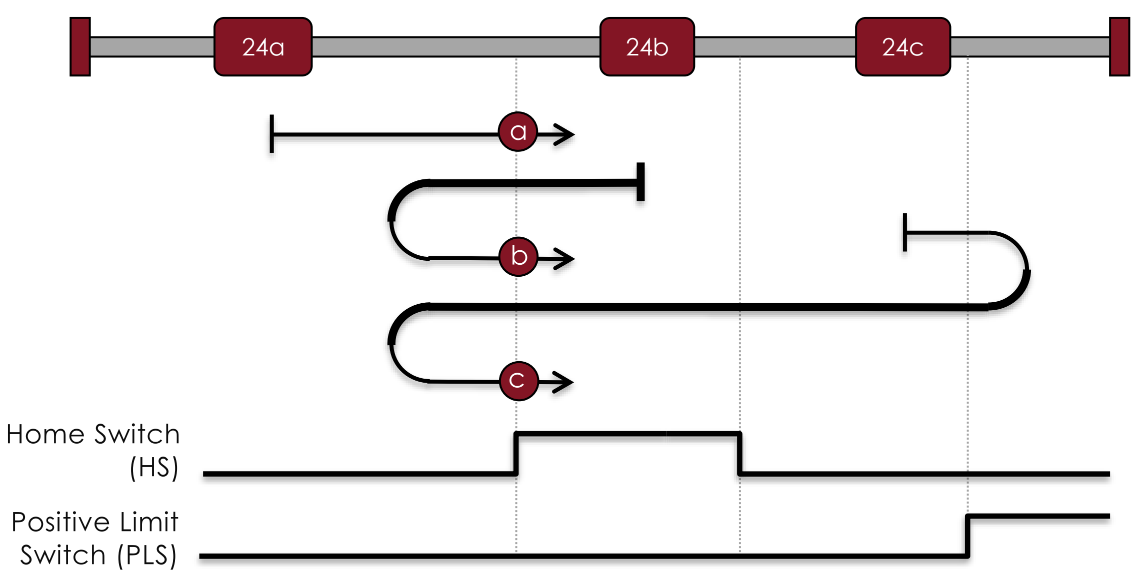


Figure 8-24: Homing Method 24

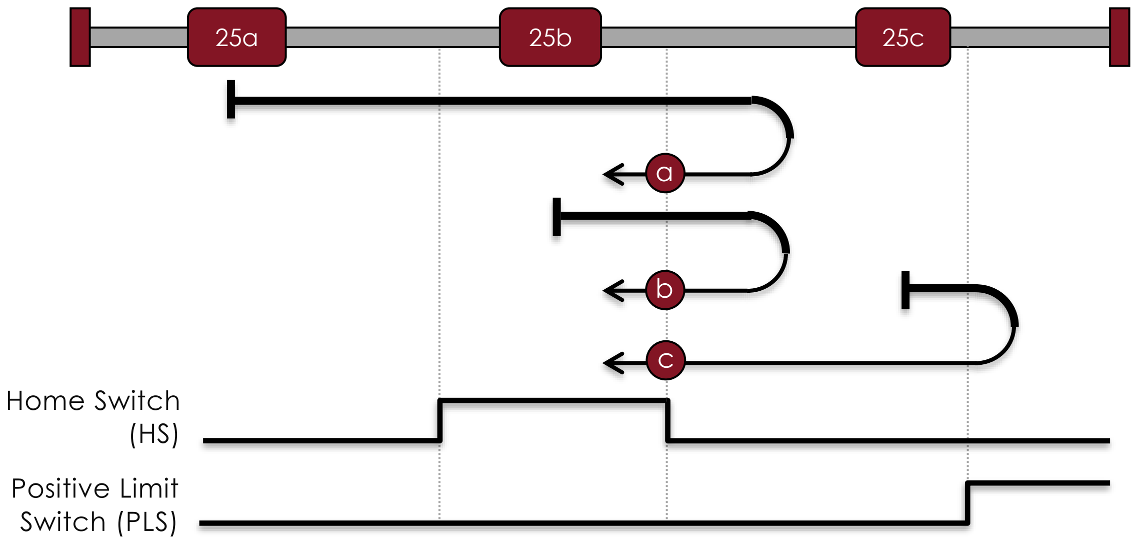


Figure 8-25: Homing Method 25

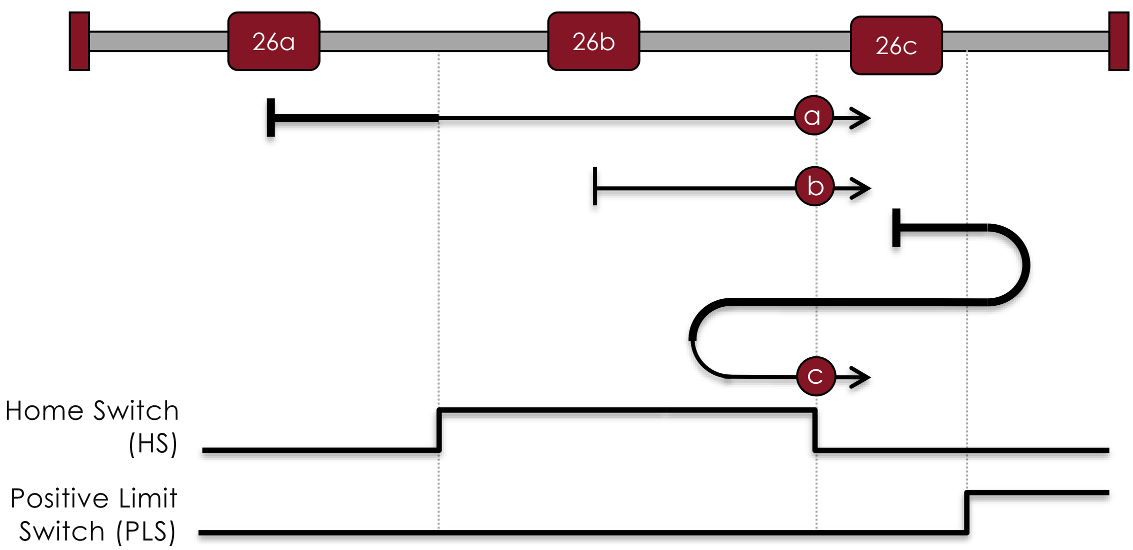


Figure 8-26: Homing Method 26

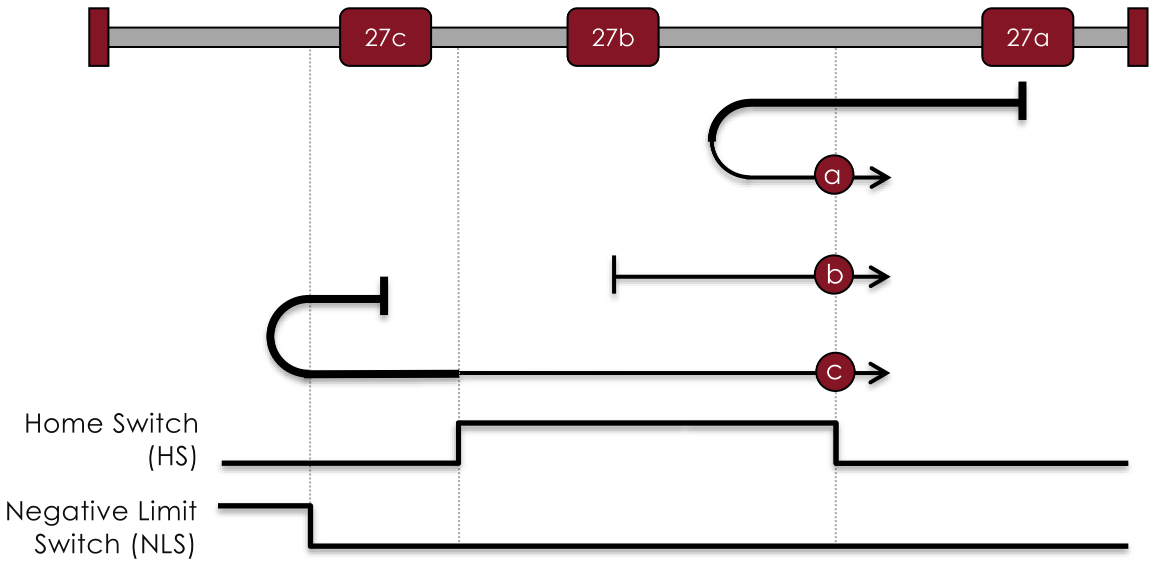


Figure 8-27: Homing Method 27

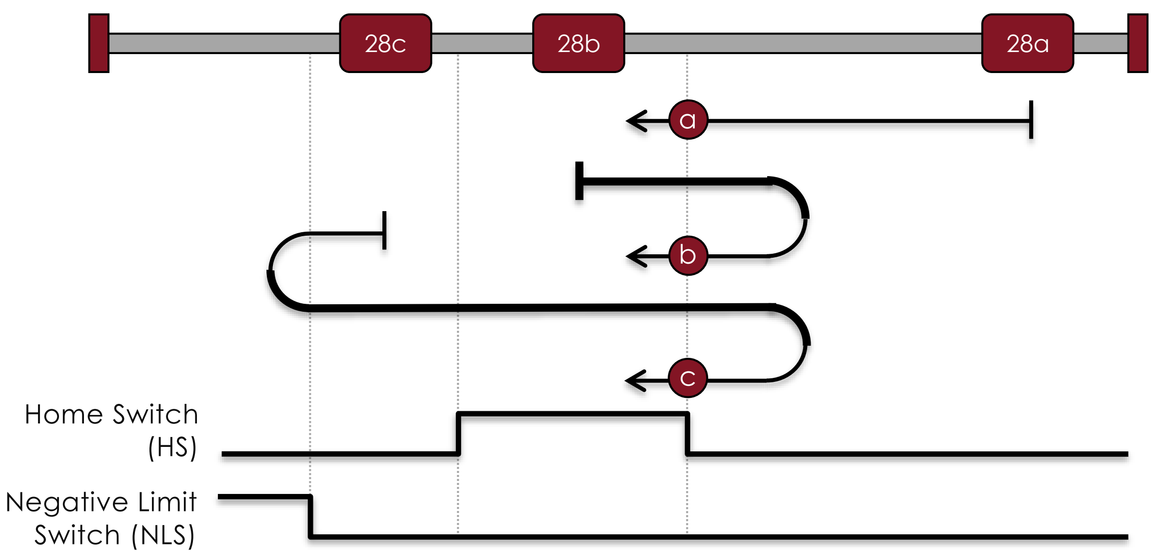


Figure 8-28: Homing Method 28

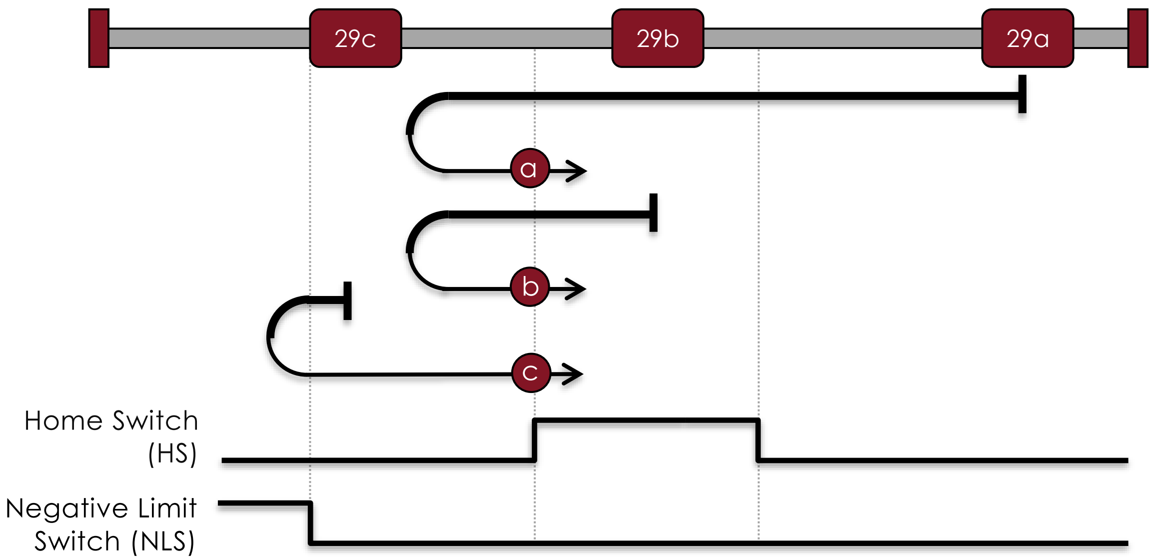


Figure 8-29: Homing Method 29

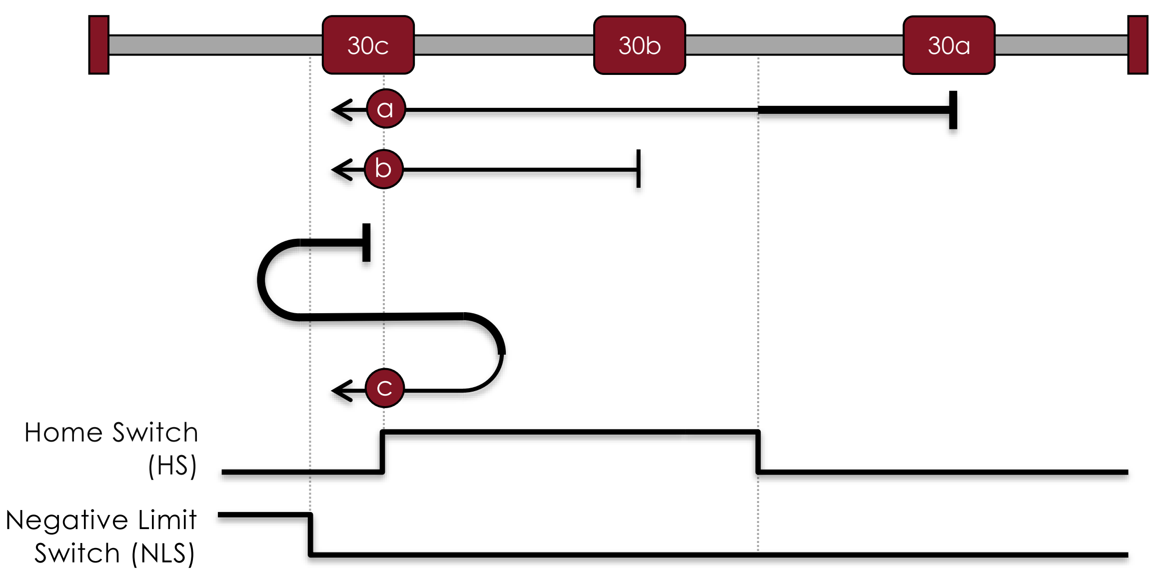


Figure 8-30: Homing Method 30

8.3.8 Method 31 and 32: Reserved

8.3.9 Method 33 and 34: Homing on Index Pulse

Using this method, the direction of homing is negative or positive respectively. The home position should be at the index pulse found in the appropriate direction as shown in Figure 8-31 and Figure 8-32.

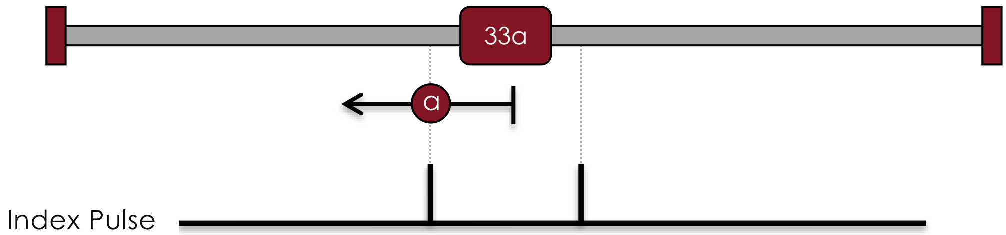


Figure 8-31: Homing Method 33

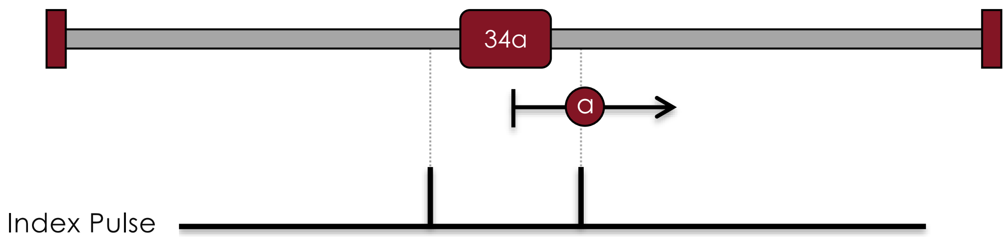


Figure 8-32: Homing Method 34

8.3.10 Method 35: Homing on the Current Position

In this method, the current position is taken to be the home position. This method does not require the drive to be in *operational enabled state.*

8.4 Use of Controlword and Statusword

The homing mode uses bits of the controlword and the statusword for mode-specific purposes. Figure 8-33 shows the structure of the controlword. Table 8-7 defines the values for bit 4 and 8 of the controlword.

15 … 9	8	7	6 … 5	4	3 … 0
(See 7.3.1)	Halt	(See 7.3.1)	Reserved (0)	Homing Operation Start	(See 7.3.1)
MSB					LSB

Figure 8-33: Controlword for Homing Mode

Table 8-7: Definition of Controlword bit 4 and bit 8

Bit	Value	Definition
4	0	Do not start homing procedure
	1	Start or continue homing procedures
8	0	Enable bit 4
	1	Stop axis according to halt option code (605Dh)

Figure 8-34 shows the structure of the statusword. Table 8-8 defines the values for bit 10, bit 12, and bit 13.

15 … 14	13	12	11	10	9 … 0
(See 7.3.2)	Homing Error	Homing Attained	(See 7.3.2)	Target Reached	(See 7.3.2)
MSB					LSB

Figure 8-34: Statusword for Homing Mode

Table 8-8: Definition of Statusword bit 10, bit 12, and bit 13

Bit 13	Bit 12	Bit 10	Definition
0	0	0	Homing procedure is in progress
0	0	1	Homing procedure is interrupted or not started
0	1	0	Homing is attained, but target is not reached
0	1	1	Homing procedure is completed successfully
1	0	0	Homing error occurred; velocity is not 0
1	0	1	Homing error occurred; velocity is 0
1	1	X	Reserved

8.5 Detailed object definitions

8.5.1 Object 607Ch: Home offset

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
607Ch	00h	Home Offset	0	No	Integer 32-bit	RW	Yes

Home offset is the amount, measured in position units, that the axis moves from the home position, attained during homing, to reach the zero position. The zero position is to be determined by the user depending on the application and set using the home offset. Once the home offsets are set and the zero position is determined, all absolute moves will then be relative to the zero position. If object is not implemented, the home offset will default to zero. This is illustrated by Figure 8-35.

Figure 8-35: Setting Zero Position Utilizing Home Offset

8.5.2 Object 6098h: Homing Method

This object indicates the homing method that should be used.

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6098h	00h	Homing Method	1	No	Integer 8-bit	RW	Yes

Table 8-9: 6098h Value Range Definition

Value	Definition
-128 to -1	Reserved
0	No homing method assigned
1 – 14	Homing combined with index pulse
15 – 16	Reserved
17 – 30	Homing without index pulse
31 – 32	Reserved
33 – 34	Homing on index pulse
35 – 127	Reserved

8.5.3 Object 6099h: Homing Speeds

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6099h	00h	Number of Elements	02h	No	Unsigned 8-bit	C	No
6099h	01h	Speed during search for switch	2000*	No	Unsigned 32-bit	RW	Yes
6099h	02h	Speed during search for zero	1000*	No	Unsigned 32-bit	RW	Yes
*The units for the value are in counts/second							

This object defines the speeds use for homing. The value is in counts/second.

8.5.4 Object 609Ah: Homing Acceleration

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
609Ah	00h	Homing Acceleration	2000*	No	Unsigned 32-bit	RW	Yes
*The units for the value are in counts/second^2							

This object defines the acceleration used during the homing process. Deceleration is performed at the same value. The value is counts/second^2.

9. Profile Position Mode

Profile position mode is currently being written. Please consult Glentek if needed.

9.1 Object 607Ah: Target Position

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
607Ah	00h	Target Position	0	No	Integer 32-bit	RW	No

This object defines the commanded position that the drive should move to in position profile mode in units of encoder counts. This object is interpreted as absolute or relative depending on “abs/rel” flag (bit 6) of Object 6040h: Controlword is set.

9.2 Object 607Bh: Position Range Limit

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
607Bh	00h	Number of Elements	2	No	Unsigned 8-bit	R	No
607Bh	01h	Min Position Range Limit	2^{31}	No	Integer 32-bit	RW	No
607Bh	02h	Max Position Range Limit	$2^{31}-1$	No	Integer 32-bit	RW	No

This object defines the maximal and minimal position range limits for Object 6062h: Position Demand Value in units of encoder counts. In the case, these limits are reached or exceeded, the input value wraps automatically to the other end of the range. Wrap- around of the input value may be prevented by setting Object 607Dh: Software Position Limit.

9.3 Object 607Dh: Software Position Limit

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
607Dh	00h	Software Position Limit	2	No	Unsigned 8 bit	R	No
607Dh	01h	Min Position Limit	2^{31}	No	Integer 32-bit	RW	Yes
607Dh	02h	Max Position Limit	$2^{31} - 1$	No	Integer 32-bit	RW	Yes

This object defines the maximal and minimal position limits for Object 6062h: Position Demand Value and Object 6064h: Position Actual Value in units of encoder counts. These limits are active only after the drive has been referenced (Homing successful).

9.4 Object 6083h: Profile Acceleration

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6083h	00h	Profile Acceleration	0	No	Unsigned 32-bit	RW	Yes
Units are in counts/s/s.							

Note: Currently the functionality of this object is not supported.

9.5 Object 6084h: Profile Deceleration

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6084h	00h	Profile Deceleration	0	No	Unsigned 32-bit	RW	Yes
Units are in counts/s/s.							

Note: Currently the functionality of this object is not supported.

9.6 Object 6085h: Quick Stop Deceleration

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6085h	00h	Quick Stop Deceleration	0	No	Unsigned 32-bit	RW	Yes

Note: Currently the functionality of this object is not supported.

9.7 Object 6086h: Motion Profile Type

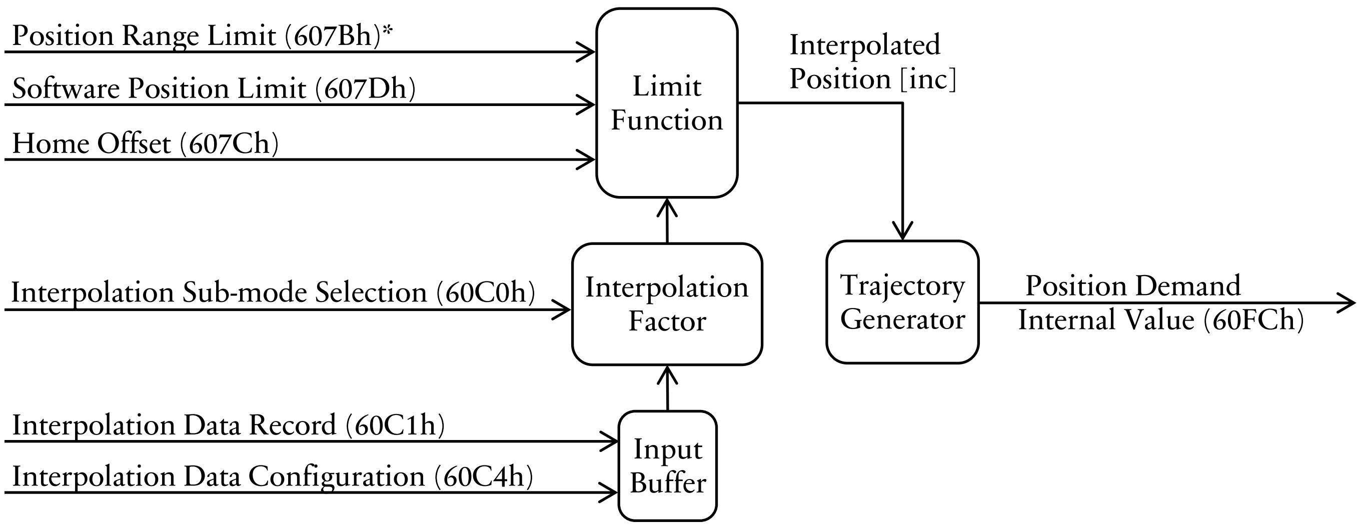
Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6086h	00h	Motion Profile Type	0	No	Integer 16-bit	RW	Yes

Note: Currently the functionality of this object is not supported.

10. Interpolation Position Mode

10.1 Overview

The interpolated position (IP) mode is used to control multiple coordinated axes or a single axis with the need for time-interpolation of set-point data. The interpolated position mode uses time synchronization mechanisms for a time coordinated of the related drive units.



*Object is in the process of being written. Please consult Glentek if needed.

Figure 10-1: Interpolation Controller

For synchronous operation, the interpolation cycle time is defined by the object interpolation time period. Time synchronization may be done by network dependent mechanisms. Each synchronization cycle actuates the next data record if a valid data record is available.

For asynchronous operation, the interpolation time (for each time slice), is included in the interpolation data record. The units for the interpolation time are specified by the interpolation time index as for synchronous operation. The next data record is actuated as the interpolation time expires and a valid data record is available. The interpolated position mode allows the controlled device to transmit a stream of interpolation data with either an implicit or explicit time reference to a drive unit.

An input buffer allows interpolation data to be sent in bursts instead of continuously in real time. The maximum size of the input buffer is read by the control device utilizing the interpolation data configuration. The buffer size is the number of interpolation data records that are sent to the drive to fill the input buffer. It is not the size in bytes.

Linear interpolation is currently only supported by Glentek's gamma drive. It requires only one interpolation data item to be buffered for the calculation of the next demand value. The interpolation data buffer in the drive is implemented as a first-in-first-out (FIFO). For each interpolation cycle, the drive calculates a position demand value by interpolating positions over a period of time.

10.2 General Definitions

For the predefined linear time interpolation, the output is a position demand internal value.

10.3 Functional Description

10.3.1 General

The next valid interpolation data record is processed in a way corresponding to the standard position mode. The standard method applies the new data immediately after the next synchronization signal in synchronous mode or after the previous interpolation time expired in asynchronous mode. An input buffer for interpolation data records eases the data exchange between control device and drive device. The real-time requirements to the network as well as to the drive device decrease in this case because an input buffer decouples the data processing in the drive device from the data transmission on the network.

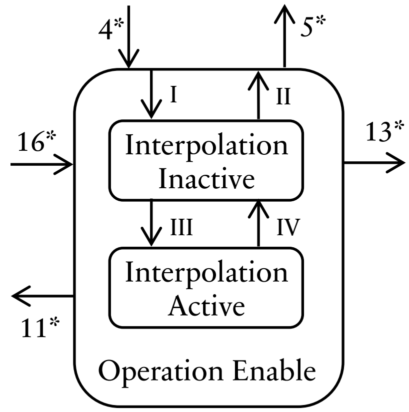
10.3.2 Buffer Strategy for Interpolated Position (IP) mode

The size of the input buffer for interpolation data records can be modified using the interpolation data configuration. After the input buffer gets organized, all previous stored data is lost. All devices supporting the interpolated position need an input buffer, which can at least keep one interpolation data record. The content of the buffer items may only be accessed via the interpolation data records.

First-in-first-out (FIFO) structure is used as input buffer. Every new received interpolation data record is placed at the end of the queue. The device takes the next data record from the top of the queue. When the last item of a data record is stored, the buffer pointer is incremented in order to point to the next buffer position. The object buffer position does not have any influence.

10.3.3 Interpolated Position Mode State Machine

Figure 10-2 specifies the interpolated position mode finite state automaton. It is a sub–State Machine of the operation enable state shown in Figure 7-2. The State Machine states support the functions shown in Table 10-1. The drive device supports the transitions and actions in Table 10-2. The event initiates the transitions. The transitions terminate after the action has been performed.



*See Figure 7-2: Drive State Machine

Figure 10-2: Interpolated Position Mode State Machine

Table 10-1: State Machine States and Supported Functions

State Machine State	Functions
Interpolation inactive	The drive device accepts input data and buffers it for interpolation calculations. Axis does not move.
Interpolation active	The drive unit accepts input data and moves the axis.

Table 10-2: Transition Events and Actions

Transition	Events	Actions
I	*ip* mode selected (See 7.4.1)	None
II	*ip* mode not selected (See 7.4.1)	None
III	Enable interpolation (bit 4 of the controlword is 1)	None
IV	Disable interpolation (bit 4 of the controlword is 0)	None

10.4 Use of Controlword and Statusword

The interpolated position mode uses some bits of the controlword and the statusword for mode-specific purposes. Figure 10-3 shows the structure of the controlword. Table 10-3 defines the values for bit 4 and bit 8 of the controlword.

15 9	8	7	6 5	4	3 0
(See 7.3.1)	Halt	(See 7.3.1)	Reserved (0)	Enable interpolation	(See 7.3.1)
MSB					LSB

Figure 10-3: Controlword for Interpolation Position Mode

Table 10-3: Definition of Controlword bit 4 and bit 8

Bit	Value	Definition
4	0	Disable interpolation
	1	Enable interpolation
8	0	Execute instruction of bit 4
	1	Stops axis according to halt option code (605Dh).

Figure 10-4 shows the structure of the statusword. Table 10-4 defines the values for bit 10 and bit 12. The *target position reached* bit remains 0 until all set-points are processed.

15 14	13	12	11	10	9 0
(See 7.3.2)	Reserved	*Ip* mode active	(See 7.3.2)	Target Reached	(See 7.3.2)
MSB					LSB

Figure 10-4: Statusword for Interpolating Position Mode

Table 10-4: Definition of Statusword bit 10 and bit 12

Bit	Value	Definition
10	0	If halt bit in last controlword was 0: Target position not yet reached If halt bit in last controlword was 1: Axle decelerates
	1	If halt bit in last controlword was 0: Target position reached If halt bit in last controlword was 1: Axle has velocity of 0
12	0	Interpolation inactive
	1	Interpolation active

10.5 Detailed Object Definitions

10.5.1 Object 60C1h: Interpolated Data Record

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60C1h	00h	Number of Elements	4	No	Integer 8-bit	C	No
60C1h	01h	1st Set-Point	0	Yes	Integer 32-bit	RW	No
60C1h	02h	2nd Set-Point	0	No	Integer 32-bit	RW	No
60C1h	03h	3rd Set-Point	0	No	Integer 32-bit	RW	No
60C1h	04h	4th Set-Point	0	No	Integer 32-bit	RW	No

This object indicates data words that are necessary to perform the interpolation algorithms. For the linear interpolation mode, each interpolation data record is regarded as a new position set-point. After the last item of an interpolation data record is written to the drive device's input buffer, the pointer of the buffer automatically increments to the next buffer position.

10.5.2 Object 60C2h: Interpolation Time Period

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60C2h	00h	Number of Elements	02h	No	Integer 8-bit	C	No
60C2h	01h	Interpolation Timer Period Value	01h	No	Unsigned 8-bit	RW	No
60C2h	02h	Interpolation Time Index	-3	No	Unsigned 8-bit	RW	No

This object indicates the configured interpolation cycle time. The interpolation time period (sub-index 01h) value is given in $10^{(\text{interpolation time index})}$ second. The interpolation time index (sub-index 02h) is dimensionless. Interpolation time period takes effect only when the interpolated mode in inactive.

Note: 60C2h Sub-Index 2 Interpolation Time Index has acceptable values 0, -1, -2, -3. Default Interpolation time period is $1 \times 10^{-3} = 1\ [ms]$

11. Profile Velocity Mode

11.1 Overview

The profile velocity mode covers the following sub-functions:

- Demand value input via trajectory generator
- Velocity capture using position sensor or velocity sensor
- Velocity control function with appropriate input and output signals
- Monitoring of the profile velocity using a window-function
- Monitoring of velocity actual value using a threshold

The operation of the reference value generator and its input parameters includes:

- Profile velocity
- Profile acceleration
- Profile deceleration
- Emergency stop
- Motion profile type

11.2 Functional Description

Figure 11-1 shows the defined structure of the profile velocity mode. The actual velocity may be obtained through differentiation from the position encoder and is represented in position encoder increments [inc/s].

The *target reached* bit (bit 10) is set to 1 in the statusword when the difference between the target velocity and the velocity actual value is within the velocity window longer than the velocity window time. As soon as the velocity actual value exceeds the velocity threshold longer than the *velocity threshold time,* then bit 12 is set to 0 in the statusword. Below this threshold, the bit is set to 1 indicating that the axis is stationary.

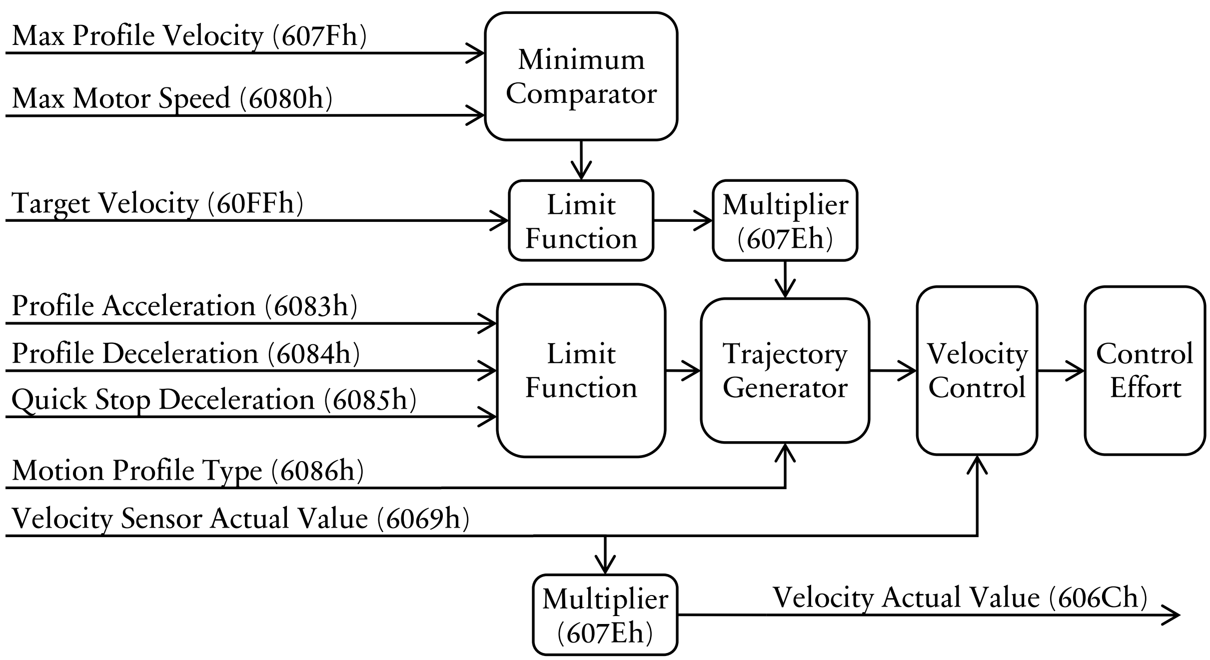


Figure 11-1: Profile Velocity Mode

11.3 Use of Controlword and Statusword

The profile velocity mode uses bit 8 of the controlword and bit 10 and bit 12 of statusword. Figure 11-2 shows the structure of the controlword. Table 11-1 defines the values for bit 8 of the controlword.

15 9	8	7	6 4	3 0
(See 7.3.1)	Halt	(See 7.3.1)	Reserved (0)	(See 7.3.1)
MSB				LSB

Figure 11-2: Controlword for Homing Mode

Table 11-1: Definition of Controlword bit 4 and bit 8

Bit	Value	Definition
8	0	The motion is executed or continued
	1	The axis will stop according to the halt option code (605Dh)

Figure 11-3 shows the structure of the statusword. Table 11-2 defines the values for bit 10 and bit 12.

15 14	13	12	11	10	9 0
(See 7.3.2)	Reserved	Speed	(See 7.3.2)	Target Reached	(See 7.3.2)
MSB					LSB

Figure 11-3: Statusword for Homing Mode

Table 11-2: Definition of Statusword bit 10 and bit 12

Bit	Value	Definition
10	0	Halt (Bit 8 in controlword) = 0; Target not reached Halt (Bit 8 in controlword) = 1; Axis decelerates
	1	Halt (Bit 8 in controlword) = 0; Target reached Halt (Bit 8 in controlword) = 1; Velocity of axis is 0
12	0	Speed is not equal 0
	1	Speed is equal 0

11.4 Detailed Object Definitions

11.4.1 Object 6069h: Velocity Sensor Actual Value

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6069h	00h	Velocity Sensor Actual Value	0	Yes	Integer 32-bit	R	No
*The units for the value are in counts/second							

This object provides the value read from a velocity sensor. The value is given in counts per second.

11.4.2 Object 606Bh: Velocity Demand Value

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
606Bh	00h	Velocity Demand Value	0	Yes	Integer 32-bit	R	No
*The units for the value are in counts/second							

This object provides the output value of the trajectory generator. The value is given in counts/second.

11.4.3 Object 606Ch: Velocity Actual Value

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
606Ch	00h	Velocity Actual Value	0	Yes	Integer 32-bit	R	No
*The units for the value are in counts/second							

This object provides the actual velocity value derived from the velocity sensor or the position sensor. The value is given in counts/second.

11.4.4 Object 606Dh: Velocity Window

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
606Dh	00h	Velocity Window	819	No	Unsigned 16-bit	RW	Yes
*The units for the value are in percentage of commanded speed							

This object defines the velocity tracking window. The value is given in percentage of commanded speed, which is used for determining if target reached status word bit 10.

Value can be found be using the formula below:

$$Value = Percentage \times 32767$$

11.4.5 Object 606Fh: Velocity Threshold

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
606Fh	00h	Velocity Threshold	1365	No	Unsigned 16-bit	RW	Yes
*The units for the value are in counts/second							

This object defines the zero-velocity threshold. The value is given in counts/second. This is used for determining if “zero” speed is reached. When “zero” speed is reached, statusword bit 12 is set to 0.

Value can be found be using the formula below:

Value = RPM * Line PPR*4/60

Example for a rotary encoder 1024 line count, 20 [rpm]:

20*1024*4/60 = 1365

11.4.6 Object 60FFh: Target Velocity

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60FFh	00h	Target Velocity	0	Yes	Integer 32-bit	RW	No
*The units for the value are in counts/second							

This object is used as target velocity input for the trajectory generator. The value is given in counts/second.

12. Profile Torque Mode

Profile torque mode is currently being written. Please consult Glentek if needed.

12.1 Object 6071h: Target Torque

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6071h	00h	Target Torque	0	No	Integer 16-bit	RW	No

Note: Currently the functionality of this object is not supported.

12.2 Object 6077h: Torque Actual Value

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6077h	00h	Torque Actual Value	0	Yes	Integer 16-bit	R	No
Units are in rated torque/1000							

Note: Currently the functionality of this object is not supported.

12.3 Object 607Fh: Max Profile Velocity

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
607Fh	00h	Max Profile Velocity	$2^{31} - 1$	No	Unsigned 32-bit	RW	Yes
Units are in counts/s							

Note: Currently the functionality of this object is not supported.

13. Cyclic Synchronous Position Mode

13.1 Overview

The overall structure for this mode is shown in Figure 13-1. With this mode, the trajectory generator is located in the control. In cyclic synchronous manner, it provides a target position to the drive, which performs position control, velocity control, and torque control. Additive velocity and torque values are provided by the control system in order to allow for velocity and/or torque feedforward. The drive device provides actual values for position, velocity, and torque to the control device.

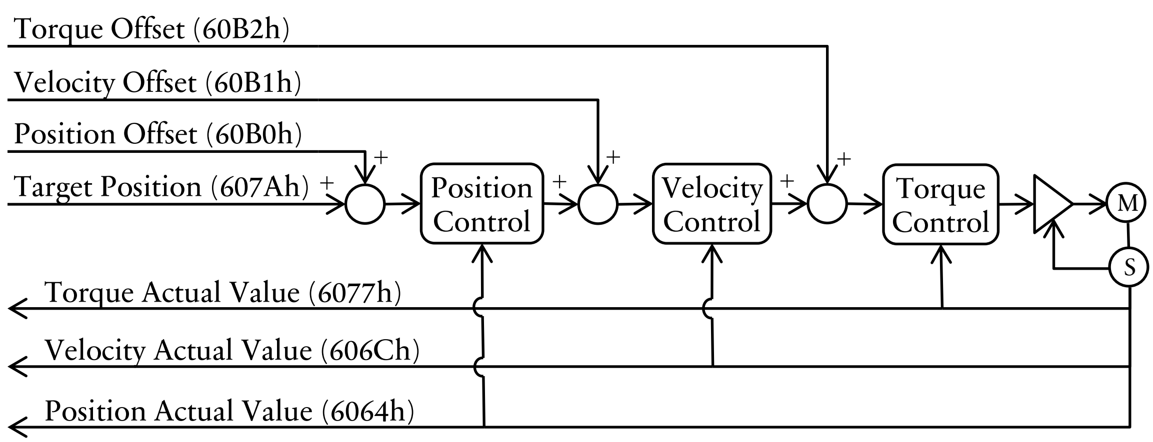
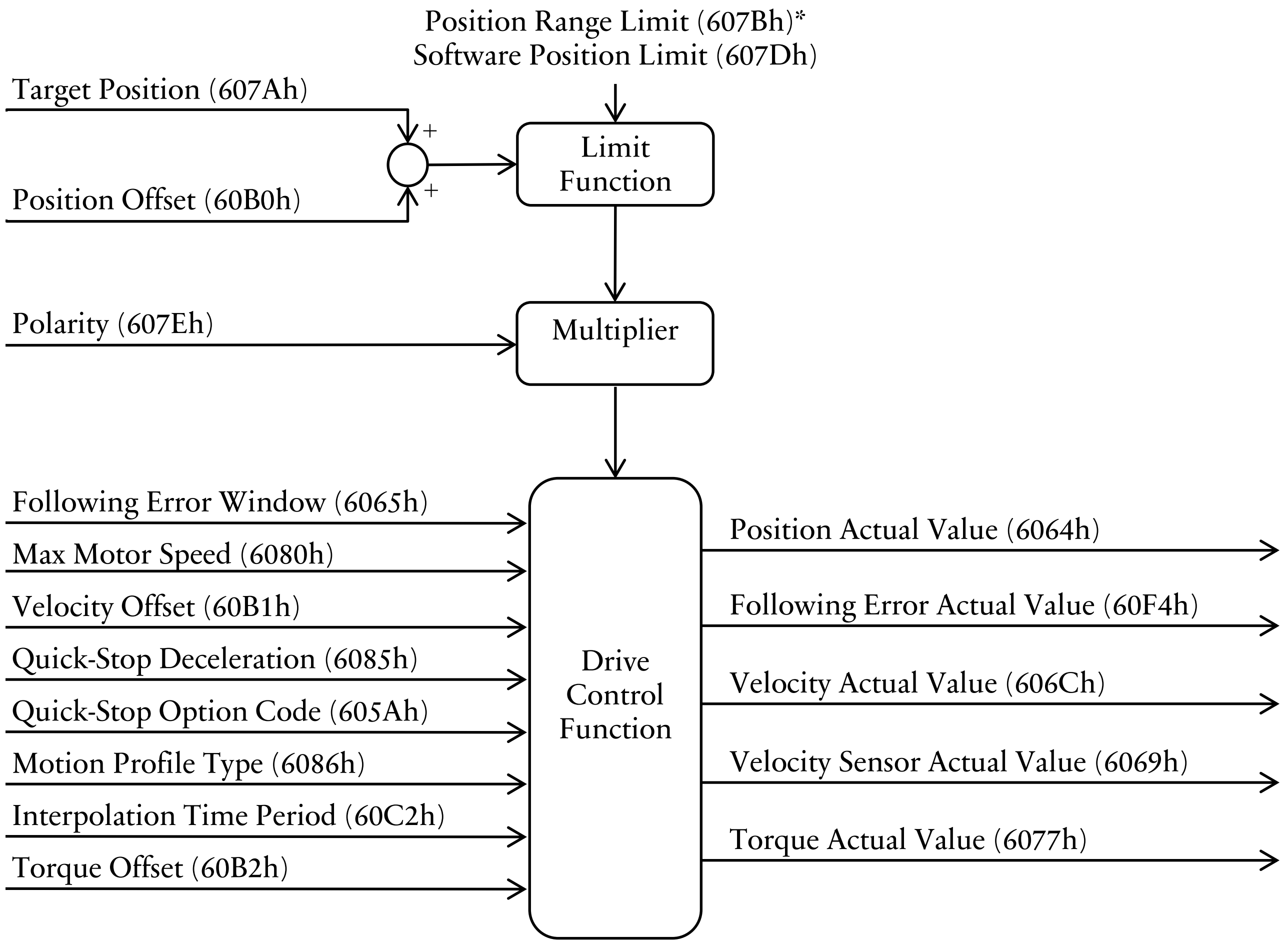


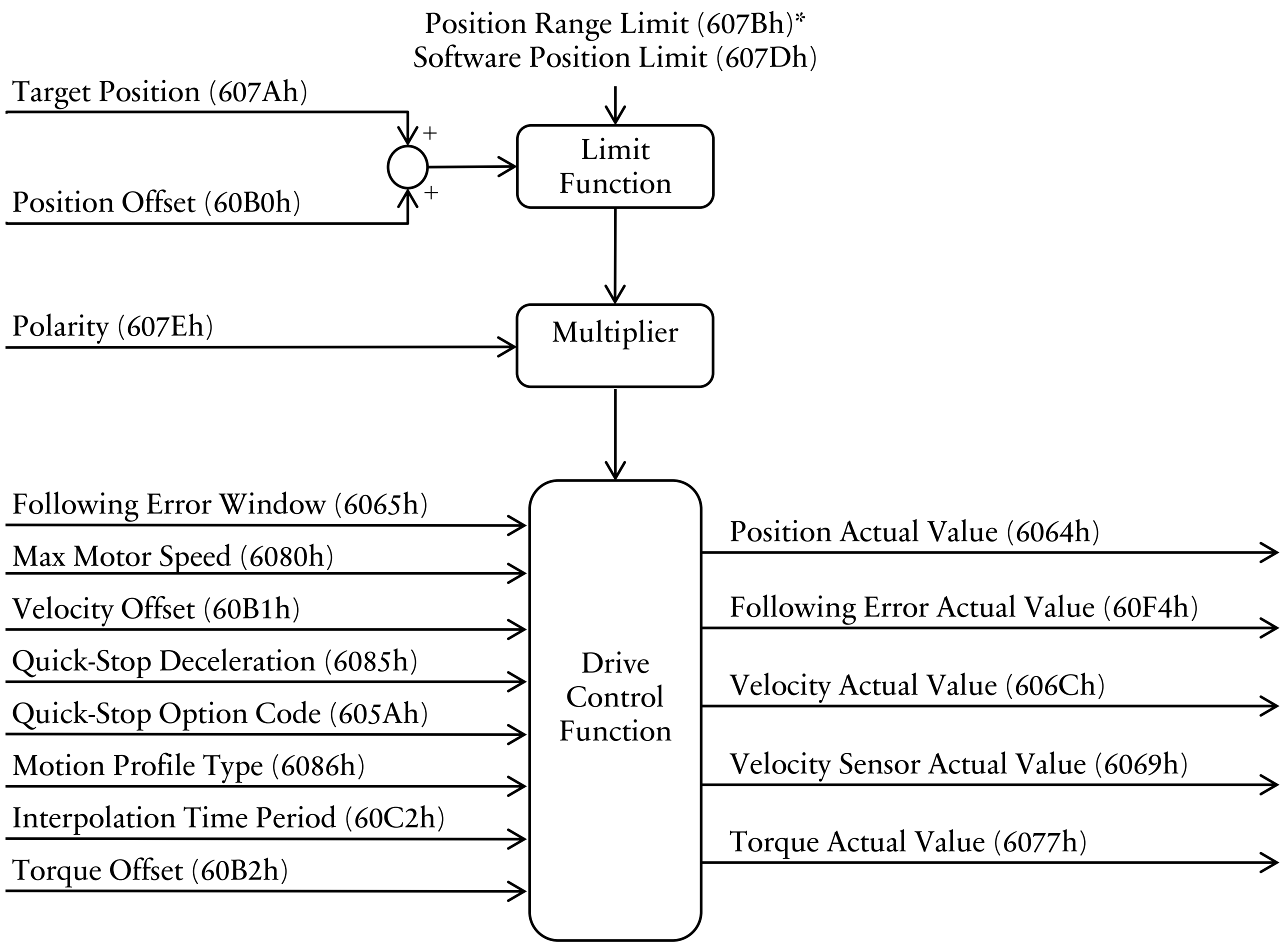
Figure 13-1: Cyclic Synchronous Position Mode Overview

13.2 Functional Description



*Object is in the process of being written. Please consult Glentek if needed.

Figure 13-2Figure 13-2: Cyclic Synchronous Position Control Function shows the inputs and outputs of the drive control function. The target position is an absolute value. The position actual value is used as an output to the control device. The velocity actual value, torque actual value, and the velocity sensor actual value are other outputs.



*Object is in the process of being written. Please consult Glentek if needed.

Figure 13-2: Cyclic Synchronous Position Control Function

13.3 Use of Controlword and Statusword

The cyclic synchronous position mode uses no mode specific bits of the controlword and three bits of the statusword for mode-specific purposes. Figure 13-3 shows the structure of the statusword. Table 13-1 defines the values of bit 12 and 13 of the statusword.

15 14	13	12	11	10	9 0
(See 7.3.2)	Following Error	Target Position Ignored	(See 7.3.2)	Reserved	(See 7.3.2)
MSB					LSB

Figure 13-3: Statusword for Profile Cyclic Synchronous Position Mode

Table 13-1: Definition of Statusword bit 12 and bit 13

Bit	Value	Definition
12	0	Target position ignored
	1	Target position is used as input to the position control loop
13	0	No following error
	1	Following error

13.4 Detailed Object Definitions

13.4.1 Object 60B0h: Position Offset

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60B0h	00h	Position Offset	0	N0	Integer 32-bit	RW	No
*The units for the value are in counts							

This object defines the offset of the target position. The offset is given in counts. The value is absolute and it is independent of how often it is transmitted over the communication system. Transmitting the value twice does not mean a double value. The additive position value is an offset to the target position, it is used to control the drive with relative values with regards to the target position.

13.4.2 Object 60B1h: Velocity Offset

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60B1h	00h	Velocity Offset	0	No	Integer 32-bit	RW	No
*The units for the value are in counts/second							

This object defines the offset for the velocity value. The offset is given in counts/second. In cyclic synchronous position mode, this object contains the input value for velocity feedforward. It contains the commanded offset of the drive device. The value is absolute and it's independent of how often it is transmitted over the communication system. Transmitting the value twice does not mean a double value. The additive velocity value is an offset to the target velocity, it is used to control the drive with relative values with regards to the target velocity.

13.4.3 Object 60B2h: Torque Offset

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60B2h	00h	Torque Offset	0	No	Integer 16-bit	R	No
*The units for the value are in drive rated current/1000							

This object defines the offset for the torque value. The offset is given in per thousand rated torque. In cyclic synchronous position mode and cyclic synchronous velocity mode, the torque offset is the input value for torque feedforward. Cyclic synchronous torque mode contains the commanded additive torque of the drive, which is added to the target torque value. The value is absolute and it is independent of how often it is transmitted over the communication system. Transmitting the value twice does not mean a double value.

14. Cyclic Synchronous Velocity Mode

14.1 Overview

The overall structure for this mode is shown in Figure 14-1. With this mode, the trajectory generator is located in the control. In cyclic synchronous manner, it provides a target velocity to the drive, which performs velocity control, and torque control. Additive velocity and torque values are provided by the control system to provide a second source for velocity and/or torque feedforward. The drive device provides actual values for position, velocity, and torque to the control device. The cyclic synchronous velocity mode covers the following sub-functions:

- Demand value input
- Velocity capture using position sensor or velocity sensor
- Velocity control function with appropriate input and output signals
- Limitation of torque demand

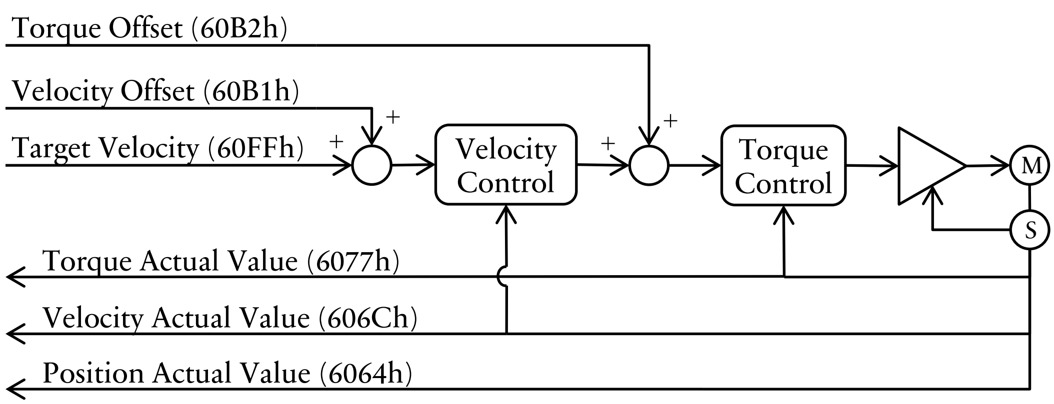


Figure 14-1: Cyclic Synchronous Position Mode Overview

14.2 Functional Description

Figure 14-2 shows the inputs and outputs of the drive control function.

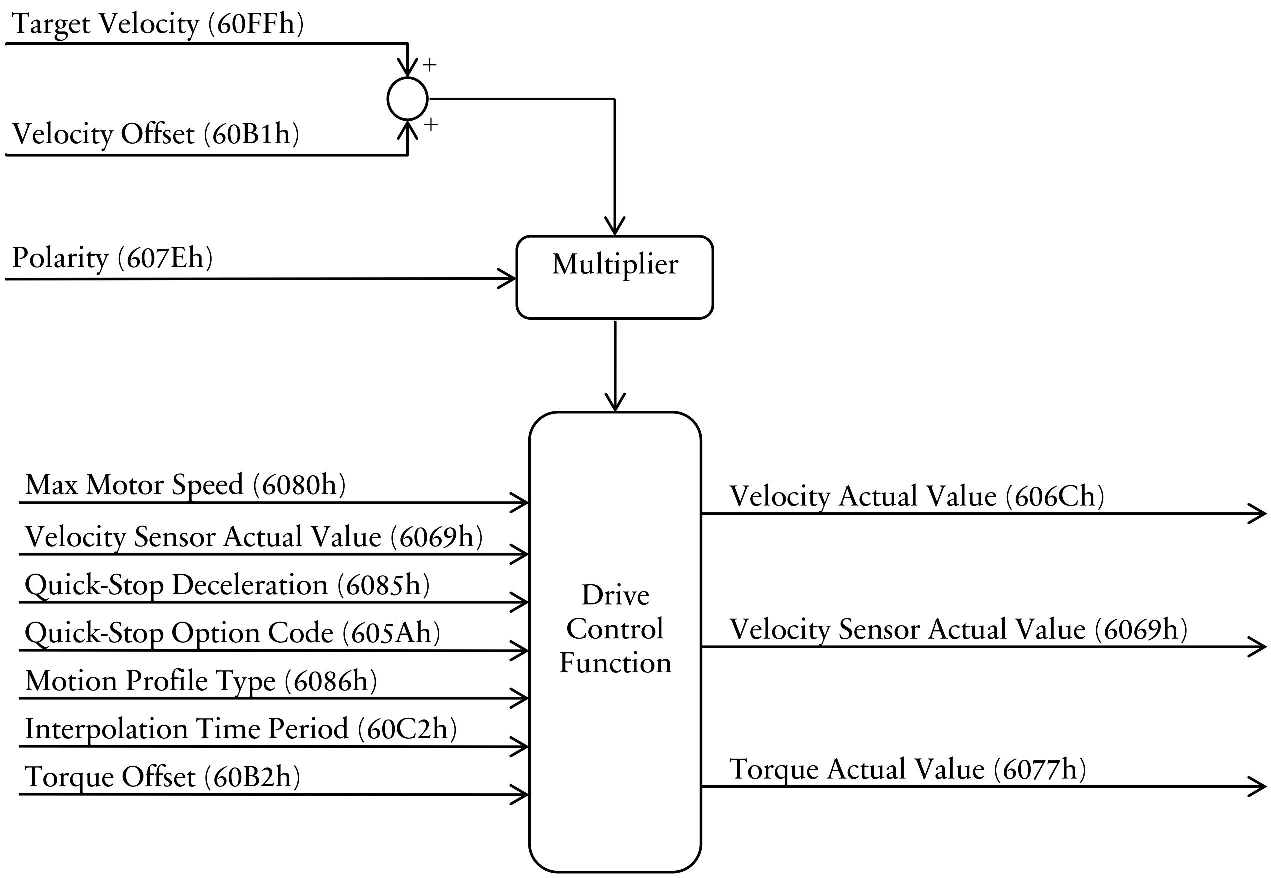


Figure 14-2: Cyclic Synchronous Velocity Control Function

14.3 Use of Controlword and Statusword

The cyclic synchronous velocity mode uses no mode specific bits of the controlword and three bits of the statusword for mode-specific purposes. Figure 14-3 shows the structure of the statusword. Table 14-1 defines the values of bit 12 of the statusword.

15 … 14	13	12	11	10	9 … 0
(See 7.3.2)	Reserved	Target Velocity Ignored	(See 7.3.2)	Reserved	(See 7.3.2)
MSB					LSB

Figure 14-3: Statusword for Profile Cyclic Synchronous Velocity Mode

Table 14-1: Definition of Statusword bit 12 and bit 13

Bit	Value	Definition
12	0	Target velocity ignored
	1	Target velocity is used as input to the velocity control loop

15. Cyclic Synchronous Torque Mode

Cyclic Synchronous Torque mode is currently being written. Please consult Glentek if needed.

16. Touch Probe Functionality

16.1 Overview

The Touch Probe function is based on IEC 61800-7-201 provides the user can capture the position information when Touch Probe position condition is reached. The position can be latched at positive or negative edge or at pulse signal of encoder.

The two touch probes can be assigned as Digital inputs by the MotionMaestro. Open **Setup** > **Setup Digital Inputs** and then assign the touch probe 1 and 2 as Figure 16-1:. Any input pins are available to assign for Touch Probe.

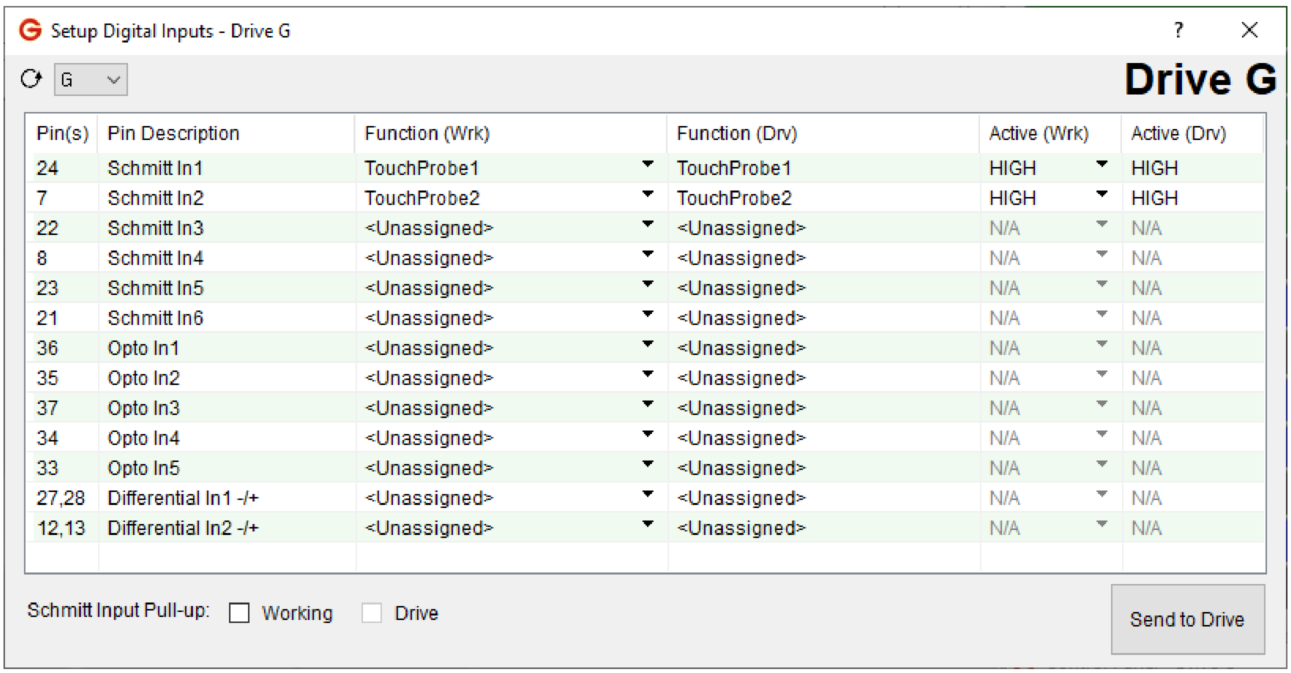


Figure 16-1: Example of Assigning Touch Probe 1 and 2

16.1.1 Object 60B8h: Touch Probe Function

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60B8h	00h	Touch Probe Function	0h	Yes	Unsigned 16-bit	RW	Yes

This object indicates the configured function of the touch probe. Table 16-1: specifies the value definition.

Table 16-1: Value Definition of 60B8h

Bit	Value	Definition
0	0	Disable touch probe 1
	1	Enable touch probe 1
1	0	Single Event Trigger
	1	Continuous Trigger
3, 2	00b	Trigger with touch probe 1 input
	01b	Trigger with zero pulse of encoder
	10b	Reserved
	11b	Reserved
4	0	Disable sampling at positive edge of touch probe 1
	1	Enable sampling at positive edge of touch probe 1
5	0	Disable sampling at negative edge of touch probe 1
	1	Enable sampling at negative edge of touch probe 1
6, 7	-	Reserved
8	0	Disable touch probe 2
	1	Enable touch probe 2
9	0	Single Event Trigger
	1	Continuous Trigger
11, 10	00b	Trigger with touch probe 2 input
	01b	Trigger with zero pulse of encoder
	10b	Reserved
	11b	Reserved
12	0	Disable sampling at positive edge of touch probe 2
	1	Enable sampling at positive edge of touch probe 2
13	0	Disable sampling at negative edge of touch probe 2
	1	Enable sampling at negative edge of touch probe 2
14, 15	-	Reserved

16.1.2 Object 60B9h: Touch Probe Status

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60B9h	00h	Touch Probe Status	0h	Yes	Unsigned 16-bit	RO	No

This object provides the status of the touch probe. Table 16-2: specifies the value definition.

Table 16-2: Value Definition of 60B9h

Bit	Value	Definition
0	0	Touch probe 1 is disabled
	1	Touch probe 1 is enabled
1	0	No touch probe 1 positive edge value captured
	1	Touch probe 1 positive edge position captured
2	0	No touch probe 1 negative edge value captured
	1	Touch probe 1 negative edge position captured
3 to 5	0	Reserved
6	0	No encoder zero pulse position captured
	1	Encoder zero pulse position captured
7	-	Reserved
8	0	Touch probe 2 is disabled
	1	Touch probe 2 is enabled
9	0	No touch probe 2 positive edge value captured
	1	Touch probe 2 positive edge position captured
10	0	No touch probe 2 negative edge value captured
	1	Touch probe 2 negative edge position captured
11 to 13	0	Reserved
14	0	No encoder zero pulse position captured
	1	Encoder zero pulse position captured
15	-	Reserved
NOTE Bit 1 and bit 2 are set to 0b when touch probe 1 is switched off (object 60B8h bit 0 is 0b). Bit 9 and 10 are set to 0b when touch probe 2 is switched off (object 60B8h bit 8 is 0b).		

16.1.3 Object 60BAh: Touch Probe 1 Positive Edge / Encoder Zero Pulse Position

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60BAh	00h	Touch Probe 1 Positive Edge / Encoder Zero Pulse Position	0h	Yes	Signed 32-bit	RO	No
The units are in counts.							

Table 16-3: Value Definition of 60BAh

60B8h	60BAh
xxxx xxxx xxx1 00x1b	Touch Probe 1 Positive Edge Captured
xxxx xxxx xxx1 01x1b	Encoder Zero Pulse Position Captured

16.1.4 Object 60BBh: Touch Probe 1 Negative Edge

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60BBh	00h	Touch Probe 1 Negative Edge	0h	Yes	Signed 32-bit	RO	No
The units are in counts.							

This object provides the position value of the touch probe 1 at negative edge. The units for the value are in counts.

16.1.5 Object 60BCh: Touch Probe 2 Positive Edge / Encoder Zero Pulse Position

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60BCh	00h	Touch Probe 2 Positive Edge / Encoder Zero Pulse Position	0h	Yes	Signed 32-bit	RO	No
The units are in counts.							

Table 16-4: Value Definition of 60BCh

60B8h	60BCh
xx1x 00x1 xxxx xxxxb	Touch Probe 2 Positive Edge Captured
xx1x 01x1 xxxx xxxxb	Encoder Zero Pulse Position Captured

16.1.6 Object 60BDh: Touch Probe 2 Negative Edge

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60BDh	00h	Touch Probe 2 Negative Edge	0h	Yes	Signed 32-bit	RO	No
The units are in counts.							

This object provides the position value of the touch probe 2 at negative edge. The units for the value are in counts.

16.2 Touch Probe Time Stamp Latch

The objects 60D1h, 60D2h, 60D3h and 60D4h provide the corresponding time stamp for touch probe 1, 2 or encoder zero pulse capture.

16.2.1 Object 60D1h: Touch Probe 1 Positive Edge Time Stamp / Encoder Zero Pulse Time Stamp

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60D1h	00h	Touch Probe 1 Positive Edge Time Stamp / Encoder Zero Pulse Time Stamp	0h	Yes	Unsigned 32-bit	RO	No
The units are in micro-seconds.							

Table 16-5: Value Definition of 60D1h

60B8h	60D1h
xxxx xxxx xxx1 00x1b	Touch Probe 1 Positive Edge Time Stamp Captured
xxxx xxxx xxx1 01x1b	Encoder Zero Pulse Time Stamp Captured

16.2.2 Object 60D2h: Touch Probe 1 Negative Edge Time Stamp

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60D2h	00h	Touch Probe 1 Negative Edge Time Stamp	0h	Yes	Unsigned 32-bit	RO	No
The units are in micro-seconds.							

This object provides the time stamp value of the touch probe 1 at negative edge. The units for the value are in micro-seconds.

16.2.3 Object 60D3h: Touch Probe 2 Positive Edge Time Stamp / Encoder Zero Pulse Position Time Stamp

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60D3h	00h	Touch Probe 2 Positive Edge Time Stamp / Encoder Zero Pulse Position Time Stamp	0h	Yes	Unsigned 32-bit	RO	No
The units are in micro-seconds.							

Table 16-6: Value Definition of 60D3h

60B8h	60D3h
xx1x 00x1 xxxx xxxxb	Touch Probe 2 Positive Edge Time Stamp Captured
xx1x 01x1 xxxx xxxxb	Encoder Zero Pulse Time Stamp Captured

16.2.4 Object 60D4h: Touch Probe 2 Negative Edge Time Stamp

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60D4h	00h	Touch Probe 2 Negative Edge Time Stamp	0h	Yes	Unsigned 32-bit	RO	No
The units are in micro-seconds.							

This object provides the time stamp value of the touch probe 2 at negative edge. The units for the value are in micro-seconds.

16.3 Touch Probe Edge Counter for Continuous Mode

For continuous touch probe mode (60B8h, bit 1 = 1b or 60B8h, bit 9 = 1b) a counter per touch probe channel is incremented on each touch probe event. Thus, the control device may check how many touch probe events happen between the control cycles. Per touch probe and per edge a counter object (objects 60D5h, 60D6h, 60D7h and 60D8h) is defined.

16.3.1 Object 60D5h: Touch Probe 1 Positive Edge Counter / Encoder Zero Pulse Counter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60D5h	00h	Touch Probe 1 Positive Edge Counter / Encoder Zero Pulse Counter	0h	Yes	Unsigned 16-bit	RO	No

Table 16-7: Value Definition of 60D5h

60B8h	60D5h
xxxx xxxx xxx1 0001b	Touch Probe 1 Positive Edge Counter Captured for single event measuring
xxxx xxxx xxx1 0101b	Encoder Zero Pulse Counter Captured for single event measuring
xxxx xxxx xxx1 0011b	Touch Probe 1 Positive Edge Counter Captured for continuous measuring
xxxx xxxx xxx1 0111b	Encoder Zero Pulse Counter Captured for continuous measuring

16.3.2 Object 60D6h: Touch Probe 1 Negative Edge Counter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60D6h	00h	Touch Probe 1 Negative Edge Counter	0h	Yes	Unsigned 16-bit	RO	No

Table 16-8: Value Definition of 60D6h

60B8h	60D6h
xxxx xxxx xxx1 0001b	Touch Probe 1 Negative Edge Counter Captured for single event measuring
xxxx xxxx xxx1 0011b	Touch Probe 1 Negative Edge Counter Captured for continuous measuring

16.3.3 Object 60D7h: Touch Probe 2 Positive Edge Counter / Encoder Zero Pulse Counter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60D7h	00h	Touch Probe 2 Positive Edge Counter / Encoder Zero Pulse Counter	0h	Yes	Unsigned 16-bit	RO	No

Table 16-9: Value Definition of 60D7h

60B8h	60D7h
xxx1 0001 xxxx xxxxb	Touch Probe 2 Positive Edge Counter Captured for single event measuring
xxx1 0101 xxxx xxxxb	Encoder Zero Pulse Counter Captured for single event measuring
xxx1 0011 xxxx xxxxb	Touch Probe 2 Positive Edge Counter Captured for continuous measuring
xxx1 0111 xxxx xxxxb	Encoder Zero Pulse Counter Captured for continuous measuring

16.3.4 Object 60D8h: Touch Probe 2 Negative Edge Counter

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60D8h	00h	Touch Probe 2 Negative Edge Counter	0h	Yes	Unsigned 16-bit	RO	No

Table 16-10: Value Definition of 60D8h

60B8h	60D8h
xxx1 0001 xxxx xxxxb	Touch Probe 2 Negative Edge Counter Captured for single event measuring
xxx1 0011 xxxx xxxxb	Touch Probe 2 Negative Edge Counter Captured for continuous measuring

16.4 Timing Diagram for Touch Probe Example

Figure 16-2: shows a timing diagram for an example touch probe configuration and the corresponding behavior. Table 16-11: explains the timing diagram.

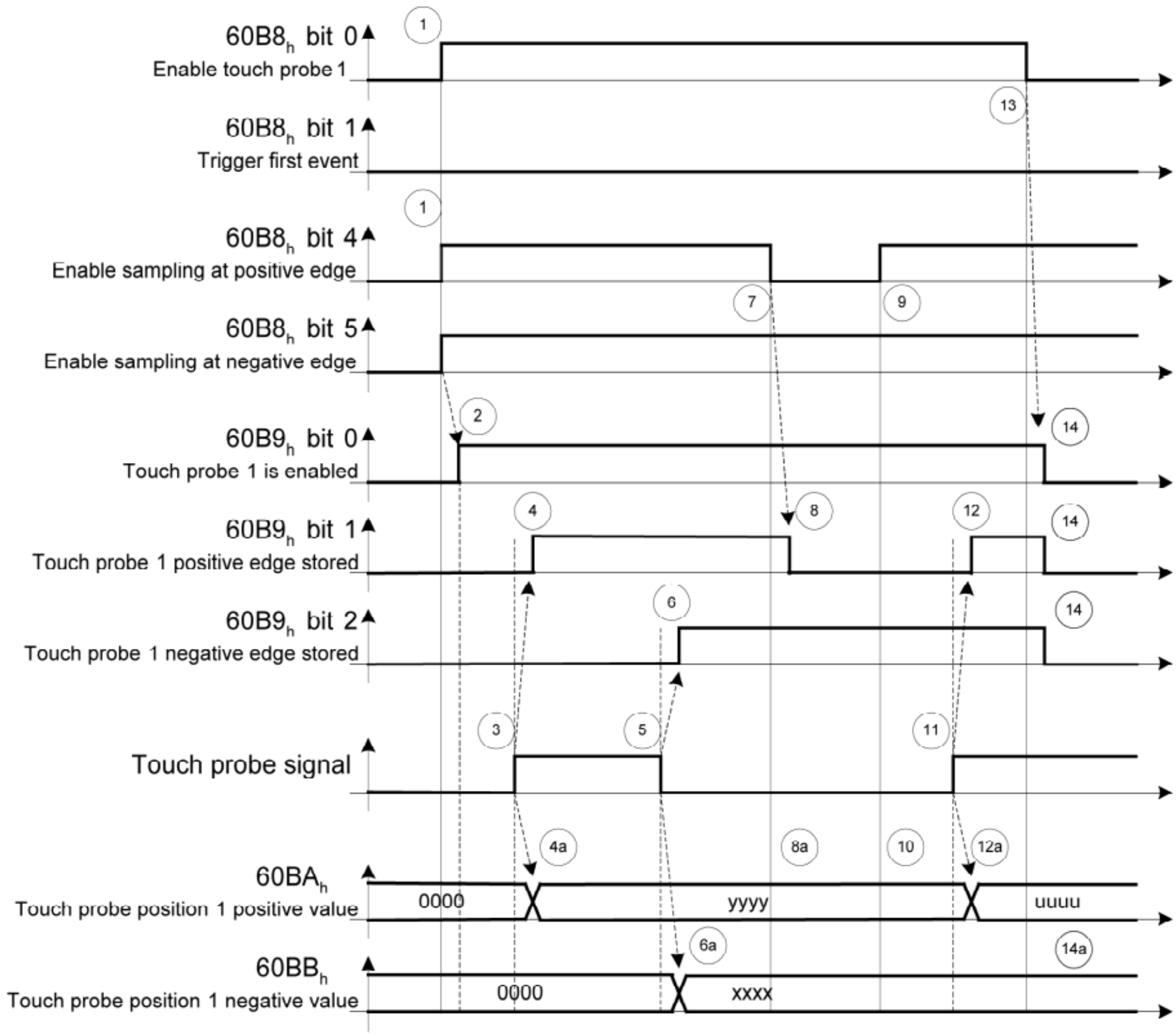


Figure 16-2: Timing Diagram for Touch Probe Example

Table 16-11: Explanation of the Timing Diagram

Number	Touch Probe Behavior
(1)	60B8h, bit 0 = 1b Enable touch probe 1 60B8h, bit 1, 4, 5 Configure and enable touch probe 1 positive and negative edge
(2)	→ 60B9h, bit 0 = 1b Status "Touch probe 1 enabled" is set
(3)	External touch probe signal has positive edge
(4)	→ 60B9h, bit 1 = 1b Status "Touch probe 1 positive edge stored" is set
(4a)	→ 60Bah Touch probe position 1 positive value is stored
(5)	External touch probe signal has negative edge
(6)	→ 60B9h, bit 2 = 1b Status "Touch probe 1 negative edge stored" is set
(6a)	→ 60BBh Touch probe position 1 negative value is stored
(7)	60B8h, bit 4 = 0b Sample positive edge is disabled
(8)	→ 60B9h, bit 0 = 0b Status "Touch probe 1 positive edge stored" is reset
(8a)	→ 60BAh Touch probe position 1 positive value is not changed
(9)	60B8h, bit 4 = 1b Sample positive edge is enabled
(10)	→ 60BAh Touch probe position 1 positive value is not changed
(11)	External touch probe signal has positive edge
(12)	→ 60B9h, bit 1 = 1b Status "Touch probe 1 positive edge stored" is set
(12a)	→ 60BAh Touch probe position 1 positive value is stored
(13)	60B8h, bit 0 = 0b Touch probe 1 is disabled
(14)	→ 60B9h, bit 0, 1, 2 = 0b Status bits are reset
(14a)	→ 60BAh, 60BBh Touch probe position 1 positive/negative value are not changed

17. Set-up and Run with Operation Mode

17.1 Set-up and Run with IP Mode

Table 17-1 shows the example of how to set up objects for Interpolation Position (IP) Mode. The drive must first be in NMT Pre-Operational Mode before setting up the objects.

Interpolation Data Record (Command) : 60C1:01h
Position Actual Value (Feedback) : 6064h
Communication Cycle period : 1000 [usec]
Interpolation time period : 1 [msec] (= 1 x 10^{-3}[sec])

Table 17-1: Example of Set-up with IP Mode

Object Index	Sub Index	Value	Size (Bits)	Description
1400h	1	80000201h	32	Disable RPDO1 (1600h)
1600h	0	00h	8	Set the number of entries in RPDO1 to zero
1600h	1	60400010h	32	Map Control Word(6040h) to 1600:01h
1600h	2	60C10120h	32	Map Position Command(60C1:01h) to 1600:02h
1600h	0	02h	8	Set the number of entries in RPDO1 to 2
1400h	2	01h	8	Set transmission type to 1 (Every Synchronous)
1400h	1	00000201h	32	Enable RPDO1 (1600h)
1800h	1	C0000181h	32	Disable TPDO1 (1A00h)
1A00h	0	00h	8	Set the number of entries in TPDO1 to zero
1A00h	1	60410010h	32	Map Status Word(6041h) to 1A00:01h
1A00h	2	60640020h	32	Map Position Actual Value(6064h) to 1A00:02h
1A00h	0	02h	8	Set the number of entries in TPDO1 to 2
1800h	2	01h	8	Set transmission type to 1 (Every Synchronous)
1800h	1	40000181h	32	Enable TPDO1 (1A00h)
1006h	0	1000	32	Set communication cycle period [usec]
60C2h	1	1	8	Set interpolation time mantissa
60C2h	2	-3	8	Set interpolation time exponent
6060h	0	7	8	Set IP mode

* Other objects for IP mode are used to default values.

After changing to NMT Operational Mode, the drive is running with IP mode.

17.2 Set-up and Run with PV Mode

Table 17-2 shows the example of how to set up objects for Profile Velocity (PV) Mode. The drive must first be in NMT Pre-Operational Mode before setting up the objects.

Target Velocity (Command) : 60FFh
Velocity Actual Value (Feedback) : 606Ch
Communication Cycle period : 1000 [usec]

Table 17-2: Example of Set-up with PV Mode

Object Index	Sub Index	Value	Size (Bits)	Description
1400h	1	80000201h	32	Disable RPDO1 (1600h)
1600h	0	00h	8	Set the number of entries in RPDO1 to zero
1600h	1	60400010h	32	Map Control Word(6040h) to 1600:01h
1600h	2	60FF0020h	32	Map Target Velocity(60FFh) to 1600:02h
1600h	0	02h	8	Set the number of entries in RPDO1 to 2
1400h	2	01h	8	Set transmission type to 1 (Every Synchronous)
1400h	1	00000201h	32	Enable RPDO1 (1600h)
1800h	1	C0000181h	32	Disable TPDO1 (1A00h)
1A00h	0	00h	8	Set the number of entries in TPDO1 to zero
1A00h	1	60410010h	32	Map Status Word(6041h) to 1A00:01h
1A00h	2	606C0020h	32	Map Velocity Actual Value(606Ch) to 1A00:02h
1A00h	0	02h	8	Set the number of entries in TPDO1 to 2
1800h	2	01h	8	Set transmission type to 1 (Every Synchronous)
1800h	1	40000181h	32	Enable TPDO1 (1A00h)
1006h	0	1000	32	Set communication cycle period [usec]
6060h	0	3	8	Set PV mode

* Other objects for PV mode are used to default values.

After changing to NMT Operational Mode, the drive is running with PV mode.

17.3 Set-up and Run with CSP Mode

Table 17-3 shows the example of how to set up objects for Cyclic Synchronous Position (CSP) Mode. The drive must first be in NMT Pre-Operational Mode before setting up the objects.

Target Position (Command) : 607Ah
Position Actual Value (Feedback) : 6064h
Communication Cycle period : 1000 [usec]

Table 17-3: Example of Set-up with CSP Mode

Object Index	Sub Index	Value	Size (Bits)	Description
1400h	1	80000201h	32	Disable RPDO1 (1600h)
1600h	0	00h	8	Set the number of entries in RPDO1 to zero
1600h	1	60400010h	32	Map Control Word(6040h) to 1600:01h
1600h	2	607A0020h	32	Map Target Position(607Ah) to 1600:02h
1600h	0	02h	8	Set the number of entries in RPDO1 to 2
1400h	2	01h	8	Set transmission type to 1 (Every Synchronous)
1400h	1	00000201h	32	Enable RPDO1 (1600h)
1800h	1	C0000181h	32	Disable TPDO1 (1A00h)
1A00h	0	00h	8	Set the number of entries in TPDO1 to zero
1A00h	1	60410010h	32	Map Status Word(6041h) to 1A00:01h
1A00h	2	60640020h	32	Map Position Actual Value(6064h) to 1A00:02h
1A00h	0	02h	8	Set the number of entries in TPDO1 to 2
1800h	2	01h	8	Set transmission type to 1 (Every Synchronous)
1800h	1	40000181h	32	Enable TPDO1 (1A00h)
1006h	0	1000	32	Set communication cycle period [usec]
6060h	0	8	8	Set CSP mode

* Other objects for CSP mode are used to default values.

After changing to NMT Operational Mode, the drive is running with CSP mode.

17.4 Set-up and Run with CSV Mode

Table 17-4 shows the example of how to set up objects for Cyclic Synchronous Velocity (CSV) Mode. The drive must first be in NMT Pre-Operational Mode before setting up the objects.

Target Velocity (Command) : 60FFh
Position Actual Value (Feedback) : 6064h
Communication Cycle period : 1000 [usec]

Table 17-4: Example of Set-up with CSV Mode

Object Index	Sub Index	Value	Size (Bits)	Description
1400h	1	80000201h	32	Disable RPDO1 (1600h)
1600h	0	00h	8	Set the number of entries in RPDO1 to zero
1600h	1	60400010h	32	Map Control Word(6040h) to 1600:01h
1600h	2	60FF0020h	32	Map Target Velocity(60FFh) to 1600:02h
1600h	0	02h	8	Set the number of entries in RPDO1 to 2
1400h	2	01h	8	Set transmission type to 1 (Every Synchronous)
1400h	1	00000201h	32	Enable RPDO1 (1600h)
1800h	1	C0000181h	32	Disable TPDO1 (1A00h)
1A00h	0	00h	8	Set the number of entries in TPDO1 to zero
1A00h	1	60410010h	32	Map Status Word(6041h) to 1A00:01h
1A00h	2	60640020h	32	Map Position Actual Value(6064h) to 1A00:02h
1A00h	0	02h	8	Set the number of entries in TPDO1 to 2
1800h	2	01h	8	Set transmission type to 1 (Every Synchronous)
1800h	1	40000181h	32	Enable TPDO1 (1A00h)
1006h	0	1000	32	Set communication cycle period [usec]
6060h	0	9	8	Set CSV mode

* Other objects for CSV mode are used to default values.

After changing to NMT Operational Mode, the drive is running with CSV mode.

18. Application Objects

18.1 Object 2000h: Motor Feedback Type

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
2000h	00h	Motor Feedback Type	-	No	Unsigned 16-bit	RO	No

This object reports the current motor feedback type.

Table 18-1: Bit Descriptions of 2000h

Bit	Description
0	TTL Encoder
1	Analog Sin/Cos Encoder
2	Resolver
3	Absolute Encoder
4	Tachometer
5	Hall Sensor
6:15	Reserved

18.2 Object 2001h: Faults of Drive

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
2001h	00h	Faults of Drive	-	Yes	Unsigned 16-bit	RO	No

This object reports the current faults of drive.

Table 18-2: Bit Descriptions of 2001h

Bit	Faults	Meaning
0	Bus Under volt	The Input Bus Voltage is below the minimum required
1	Commutation Init	Hall angles do not match encoder counter angle. No Halls: Phase finding routine failed
2	Internal Fault	An Internal Fault has occurred
3	Reserved	-
4	EEPROM	EEPROM checksum fault – Contact a Glentek application engineer
5	Hall	The drive has detected an invalid Hall state on the Hall inputs from the motor
6	Motor Over Temp	The drive has detected a motor over temp signal on the *Motor Overtemp* Input or on the I/O Connector
7	Reserved	-
8	RX Overflow	Software overflow
9	Position Following Error	Position Mode Only – Difference between commanded position and actual position is excessive
10	Over Speed	The Motor speed has exceeded the specified threshold
11	Encoder	The drive has detected a fault on one or more of the encoder inputs
12	Drive Over Temp	The drive has exceeded the drive maximum allowed temperature
13	Bus Over Volt	The Input Bus Voltage to drive has exceeded the allowed max voltage
14	LSECB (Low Speed Electronic Circuit Breaker)	The drive output current has exceeded the time + current constraints programmed into the drive
15	HSECB (High Speed Electronic Circuit Breaker)	The drive has detected a short circuit on one or more of the output terminals

18.3 Object 2002h: Status of Drive

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
2002h	00h	Status of Drive	-	Yes	Unsigned 16-bit	RO	No

This object reports the current status of drive.

Table 18-3: Bit Descriptions of 2002h

Bit	Status	Meaning
0	STO Active	STO (Safe Torque Off) input activated
1	Drive Fault	One or more of the internal drive faults has occurred
2	Commutation Init	The drive is finding the encoder commutation Tracks/Hall signals and calculating the rotor angle
3	Software Inhibit	The drive is disabled from within MotionMaestro software
4	At Zero	The motor is at zero rpm
5	At Speed	The motor is at a specified rpm
6	Drive Commutated	Commutation Initialization was successful
7	Encoder Index Sensed	The drive has detected the encoder index signal
8	Drive Enabled	The drive is not inhibited and no faults are present
9	Hardware Inhibit	An inhibit signal has been received at one of the I/O terminals
10	Current Fold back	The fold back function has occurred, motor current has been reduced to the configured level
11	Balancing	The drive is in the process of balancing the motor phase currents
12	Stopping	The drive is decelerating the motor
13	External Reset	The reset terminal of the I/O has been activated from an external source
14	Auto Phasing	The drive is in auto phasing
15	Reserved	-

18.4 Object 2009h: Absolute Torque Actual Value

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
2009h	00h	Absolute Torque Actual Value	0h	Yes	Unsigned 16-bit	RO	No

Note: This object is currently being implemented. Please consult Glentek if needed.

18.5 Object 6062h: Position Demand Value

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6062h	00h	Position Demand Value	0	Yes	Integer 32-bit	R	No
Units are in Counts.							

This object shows the motor position which the drive is attempting to move the axis in units of encoder counts.

18.6 Object 6063h: Position Actual Internal Value

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6063h	00h	Position Actual Internal Value	0	Yes	Integer 32-bit	R	No
Units are in Counts.							

This object shows the actual motor position calculated by the drive-in units of encoder counts.

18.7 Object 6064h: Position Actual Value

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6064h	00h	Position Actual Value	0	Yes	Integer 32-bit	R	No
Units are in Counts.							

This object shows the actual value of motor position in units of encoder counts.

18.8 Object 6065h: Following Error Window

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6065h	00h	Following Error Window	0	No	Integer 32-bit	RW	Yes

This object defines the configured range of tolerated position values symmetrically to the position demand value in units of encoder counts. In the case actual position value is out of the following error window, a following error results. A following error may occur when a drive is block, unreachable profile velocity occurs, or at wrong closed-loop coefficients. If the value of the following error wind is FFFFFFFFh, the following control shall be switched off.

18.9 Object 6067h: Position Window

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6067h	00h	Position Window	32h	No	Unsigned 32-bit	RW	Yes
Units are in Counts.							

This object defines the configured range of tolerated position values symmetrically to Target Position in units of encoder counts. If the actual value of the position encoder is within the position window, the target position shall be regarded as having been reached.

18.10 Object 6072h: Max Torque

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6072h	00h	Max Torque	03E8h	Yes	Unsigned 16-bit	RW	Yes
The units are in drive rated torque/1000.							

The default value 1000(=03E8h) means 100.0% of drive rated torque. The default value is determined by setting 'Current Limit' on MotionMaestro.

18.11 Object 607Eh: Polarity

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
607Eh	00h	Polarity	0	No	Unsigned 8-bit	RW	Yes

This object indicates if the position demand value shall be multiplied by 1 of by -1. The polarity flag has no influence on homing mode. The Position Polarity bit is used for profile position and cyclic synchronous position mode. The velocity polarity bit is used for profile velocity and cyclic synchronous velocity mode. Polarity in a position or velocity profile is chosen according to Table 18-4.

7	6	5		0
Position Polarity	Velocity Polarity	Reserved (0)		
MSB				LSB

Figure 18-1: Statusword for Profile Cyclic Synchronous Position Mode

Table 18-4: Definition of Bits 6 and 7

Bit	Value	Definition
6	0	Multiply by 1
	1	Multiply by -1
7	0	Multiply by 1
	1	Multiply by -1

18.12 Object 6080h: Max Motor Speed

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
6080h	00h	Max Motor Speed	$2^{31}-1$	No	Unsigned 32-bit	RW	Yes
Units are in counts/s							

This object indicates the configured maximal allowed speed for the motor in either direction.

18.13 Object 60F4h: Following Error Actual Value

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60F4h	00h	Following Error Actual Value	0	Yes	Integer 32-bit	R	No
Units are in counts.							

This object reports the difference: Object 6062h: Position Demand Value - Object 6064h: Position Actual Value.

18.14 Object 60FCh: Position Demand Internal Value

Object	Sub-Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60FCh	00h	Position Demand Internal Value	0	Yes	Integer 32-bit	R	No
Units are in Counts.							

This object reports the position calculated by the motion profile in units of encoder counts; it takes into the account the acceleration and velocity targets.

18.15 Object 60FDh: Digital Inputs

Object	Sub Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60FDh	00h	Digital Inputs	0	No	Unsigned 32-bit	R	No

The object 60FDh: Digital Inputs can be used after assigning of the programmable digital input by MotionMaestro.

Table 18-5: Definition of 60FDh

Bit	Definition
0	Reserved
1	External Inhibit
2	External Reset
3	Positive Limit
4	Negative Limit
5	Fault Input
6	Stop/Go Control
7	E-Stop
8	Aux Encoder Fault
9	Drive Operation Control
10	Profile Generator Trigger Bit
11	Profile Generator Stop Bit
12	Set Home Bit
13	Safe Stop1
14-19	Reserved
20	Touch Probe 1 Input
21	Touch Probe 2 Input
22-31	Reserved

18.16 Object 60FEh: Digital Outputs

Object	Sub Index	Description	Default	PDO Map	Data Type	Access	Save to EEPROM
60FEh	00h	Digital Outputs	2h	No	Unsigned 16-bit	RO	No
60FEh	01h	Digital Outputs 1	0h	Yes	Unsigned 32-bit	RW	No
60FEh	02h	Reserved	0h	Yes	Unsigned 32-bit	RW	No

The object 60FE:01h Digital Outputs 1 can be used after assigning of the programmable digital output by MotionMaestro.

Table 18-6: Definition of 60FE:01h Digital Outputs 1

Bit	Definition
0	Digital Out: Bit1
1	Digital Out: Bit2
2	Digital Out: Bit3
3	Digital Out: Bit4
4	Digital Out: Bit5
5	Digital Out: Bit6
6	Digital Out: Bit7
7	Digital Out: Bit8
8-31	Reserved

For example, Open **Setup** > **Setup Digital Outputs** at MotionMaestro and then assign Digital Out: Bit 1 as Figure 18-2.

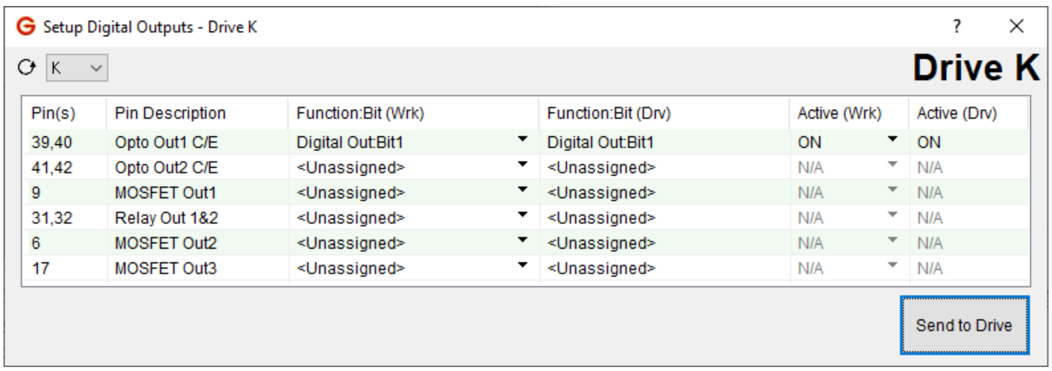


Figure 18-2: Example of Assigning Digital Outputs

Appendix 1. Object Reference

A.1. Communication Profile

A.2. Drive and Motion Control Profile

Appendix 2. References

[1] CiA 301 CANopen application layer and communication profile

Glentek's CANopen® Communication

Communication Manual

Revision 02

www.glentek.com

208 Standard St.

El Segundo, CA 90245

T: (310) 322-3026

F: (310)322-7709