
# CANopen COMMUNICATION MANUAL

300 Series CANopen Drives

Rev 3.0.3a

Revision History

| Revision | Date | Comments |
|---|---|---|
| 1.0 | Feb 17, 2006 | First draft |
| | | |

# Communication Manual

### 1.1.1 Purpose of this manual

This manual will provide all information necessary to communicate with and operate Advanced Motion Controls CANopen drives. Further information regarding the physical CAN layer and CANopen protocol is attainable through the DSP402 and DS301 documentation.

The CAN interface for the Advanced Motion Controls' digital drives follows the CiA DS301 communications profile and the CiA DSP402 device profile (device profile for drives and motion control). CiA (CAN in Automation) is the non-profit organization that governs the CANopen standard. They can be contacted at http://www.can-cia.org.

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 descriptions, while CANopen refers to the communication protocol.

### 1.1.2 Differences between this manual and DS301 & DSP402

This manual provides all information necessary to properly communicate with the drive via the CANopen interface. The DS301 and DSP402 documents are complimentary and can be used if more detailed information is required on specific standard CANopen features.

# 2 CANOPEN OBJECTS

Every AMC CANopen drive function is defined by groups of objects. An object is roughly equivalent to a memory location that holds a value. The values stored in the drive's objects are used to perform the drive functions (current loop, velocity loop, position loop, I/O functions, etc.).

The drive has a unique object for every parameter that needs to be stored or used. Access to the objects varies depending on what the object is used for. Objects may be writable, readable, or both. Some objects are state dependant such that they may only be written to if the drive is in a certain state (e.g. disabled state). The list of objects that AMC CANopen servo drives use is found in the Object Dictionary. Each table in the object dictionary describes the important information regarding that object including: object index, sub-indices, units, and accessibility.

Each object is accessible with a 16-bit address called the object index. Some objects contain sub components with 8-bit addresses called sub indices. Reading and writing to objects is accomplished via CANopen Messages. Specific types of messages are designed to access specific objects. Details about CANopen message types are found in the CANopen Messages.

## 2.1 Types of CANopen Objects

There are 3 main object categories:

### *2.1.1 Communication Objects 1000h – 1FFFh*

These objects relate to CANopen communication, more specifically they relate to objects defined by the DS301 communication profile. Objects in this range are used to configure CANopen messages (see CANopen Message Structure) and general CANopen network settings (e.g. network watchdog).

### *2.1.2 Manufacturer Specific Objects 2000h – 5FFFh*

These objects are manufacturer specific. Detailed information about the AMC manufacturer specific objects can be found in this manual and in the Object Dictionary.

### *2.1.3 Standard Servo Drive Objects 6000h – 9FFFh*

These objects are the standardized device profile objects. Objects in this range relate to the device profile of the CANopen device. The applicable device profile for AMC CANopen drives is DSP402 (CANopen profile for servo drives). Other device profiles exist also, but they are not discussed here; examples include: DS401 (CANopen profile for I/O modules), and DS405 (CANopen profile for PLC). Detailed information about the DSP402 objects supported by the AMC CANopen drives can be found in this manual.

## 2.2 CANopen Object Data

Every CANopen object index - and sub-index if available - is an address pointer to a data location. The 16-bit index and 8-bit sub index make it effectively a 24-bit address space. The data type can be any type typically found in digital systems, such as 8-bit, 16-bit, 32-bit, string… The data type can also be a record (in the case of an index with sub-indices), with multiple record entries, and each entry can be of the above mentioned data type. Nested records are not allowed.

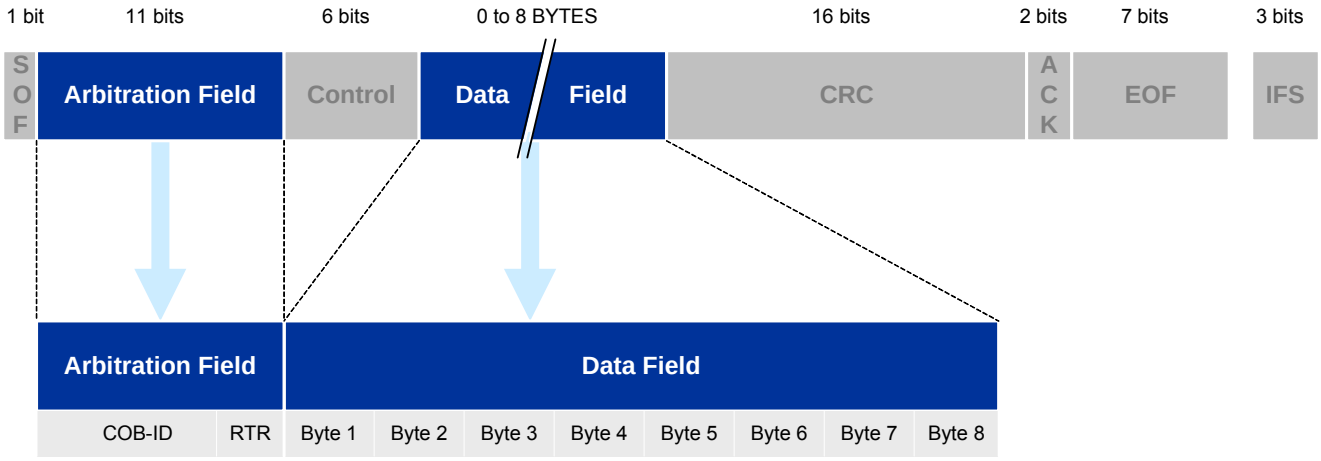
# 3 CANopen Message Structure

CANopen messages exchange information between the CANopen host (master) and the CANopen nodes (slave). When collecting information, a host may either poll, or simply wait, for important messages in the network.

Although the host may gather information through “polling” (i.e. the host continuously requesting information updates from each node), a more effective method is to exchange information in an interrupt driven fashion (i.e. information is exchanged only when there is new information available). Both mechanisms are possible within the CANopen framework, but the interrupt driven exchange method requires much less overhead, thus allowing higher data throughput.

Most messages either read or write data to objects contained in the network nodes. There are 8 types of messages used in a CANopen system. Each message type gets a detailed explanation in CANopen Messages. Regardless of message type, the general structure of a CANopen message is the same. CANopen messages fit within one CAN frame where there are only two parts of the CAN frame the user needs to access, namely the Arbitration, and Data fields. All other fields are automatically configured by the CAN hardware.

**Figure 1: CANopen frame bit sequence**




## 3.1 The Arbitration Field

The values in the arbitration field set the priority of the message. The closer the value is to 0h, the higher the priority of the message. Higher priority messages will dominate, or take precedence, over other messages on the CAN bus. Arbitration of the CAN bus is done at the CAN hardware level, thus ensuring that the highest priority message is transmitted first.

CANopen message priority is determined by the message COB-ID bits and the RTR (Remote Transmit Request) bit. Within the CANopen framework, there are 7 COB-ID ranges. One COB-ID range is used twice, resulting in 8 message types. Each message type is described in detail in CANopen Messages.

**Figure 2: Arbitration field values.**

| Arbitration Field | | Data Field | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| COB-ID | RTR | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
| 11-bit Identifier | 1 or 0 | xx | xx | xx | xx | xx | xx | xx | xx |

### *3.1.1 COB-ID*

Every CANopen message has a unique COB-ID that identifies the message type and in case of node specific messages, the node number. Table 1 contains the COB-ID or COB-ID range for each message type. In the

case of a range of COB-IDs, the actual COB-ID for a message will depend on which node receives or transmits the message. These COB-IDs begin with a base number (assigned in CiA's DS301 specification) and the addition of the NODE-ID completes the COB-ID. If the COB-ID field base is 600h, for example, a COB-ID of 605h pertains to a message (of type SDO as per table 2 below) to/from node 5 in the CANopen network. Each message type is described in detail in CANopen Messages.

**Table 1: CANopen message types**

| Message Type | Description | COB-ID |
|---|---|---|
| NMT | Network Management (broadcast) | 0h |
| NMT Error Control | Network management error control | 701h – 77Fh |
| BOOT-UP | Boot-Up message | 701h – 77Fh |
| SYNC | Synchronization message (broadcast) | 80h |
| EMERGENCY | Emergency messages | 81h - FFh |
| TIME STAMP | Time stamp (broadcast) | 100h |
| PDO | Process Data Objects | 181h - 57Fh |
| SDO | Service Data Objects | 581h – 67Fh |

### *3.1.2 RTR Bit*

The remote transmission request (RTR) bit is used in some specific cases when the host would like to request information from a node. In particular, the RTR bit is used for node guard and TPDO requests. With the exception of these two cases, the RTR bit is always set to 0.

### *3.1.3 Node-ID*

Every node on the CANopen network must have a unique node-ID, between 1 and 127. Node 0 is always considered the host. See the hardware manual for configuration of the drive node-ID.

## 3.2 The Data Field

The content of the Data field depends on the CANopen message type. Detailed information about the CANopen message data is found under the appropriate message type in CANopen Messages while details on each object are found in the Object Dictionary.

### *3.2.1 Little Endian Format*

Numerical data larger than 1 byte must be organized into "Little Endian" format. This means that the data is broken into its individual bytes and sent Least-Significant-Byte-First. The 24-bit number 102315h, for example, must be transmitted LSB (Least Significant Byte) first as 15h 23h 10h (as shown in Figure 3 below).

**Figure 3: Sending 102315h in Little Endian format**

| Arbitration Field | | Data Field | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| COB-ID | RTR | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
| AAAh | X | 15h | 23h | 10h | 00h | 00h | 00h | 00h | 00h |

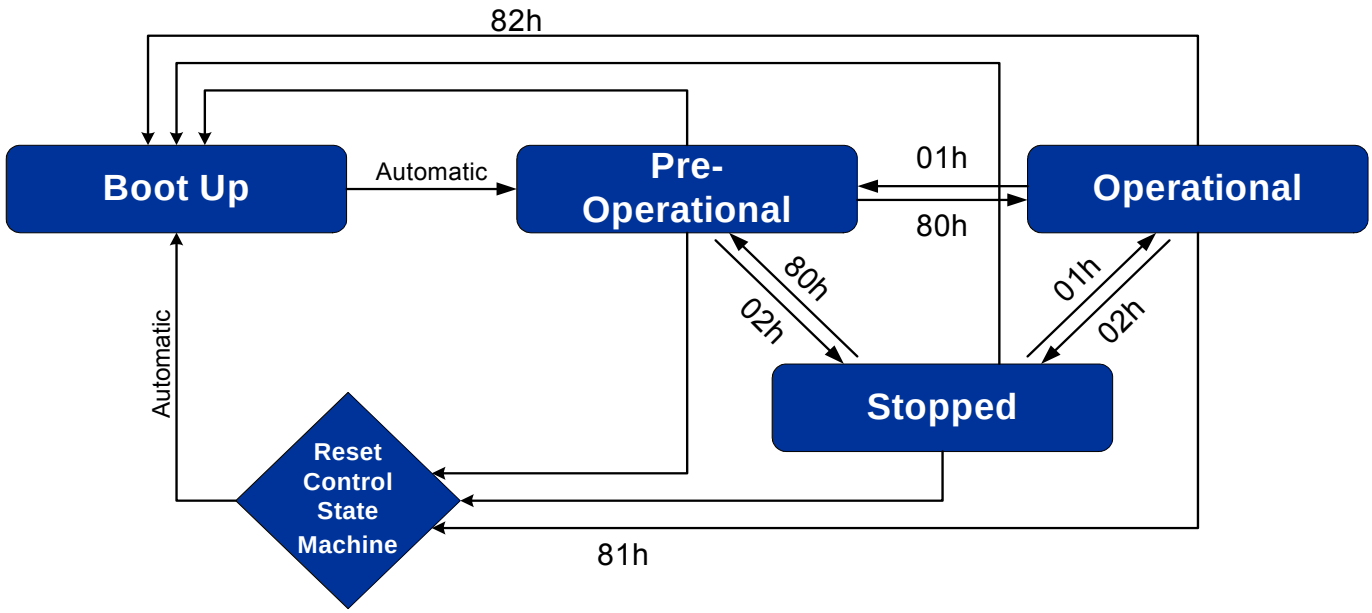
## 3.3 CAN Bus Traffic Concerns

It is best to keep the network idle for at least 50% of the time (50% bus load). Busload will depend on CAN bus bit rate and CANopen message rates.

AMC CANopen drives support 8 message types. Each message type fits within the defined structure of a CAN frame. The data field of each message type can vary, but all messages require the arbitration field to be populated with the appropriate COB-ID. NMT service, SYNC, and TIME STAMP messages have fixed COB-ID's while the other message types use a range of values.

## 4.1 NMT Messages

**Figure 4: Communication State Machine operation.**




Every CANopen device contains an internal Network Management server that communicates with an external NMT master. One device in a network, generally the host, may act as the NMT master. Through NMT messages, each CANopen device's network management server controls state changes within its built-in Communication State Machine. This is independent from each node's operational state machine, which is device dependant and described in Control State Machine. It is important to distinguish a CANopen device's operational state machine from its Communication State Machine. CANopen sensors and I/O modules, for example, have completely different operational state machines than servo drives. The "Communication State Machine" in all CANopen devices, however, is identical as specified by the DS301.

NMT messages have the highest priority. The 5 NMT messages that control the Communication State Machine each contain 2 data bytes that identify the node number and a command to that node's state machine. Table 2 shows the 5 NMT messages supported by AMC, and Figure 5 shows the correct message construction for sending these messages.

**Figure 5: NMT message construction**

| Arbitration Field | | Data Field | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| COB-ID | RTR | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
| 000h | 0 | See Table 2 | See Table 2 | These bytes not sent | | | | | |

Table 2: NMT messages supported by AMC CANopen servo drives.

| NMT Message | COB-ID | Data Bytes | | Description |
|---|---|---|---|---|
| | | 1 | 2 | |
| Start Remote Node | 0 | 01h | Node-ID* | Sets the CANopen communication state machine on the designated node to Operational. |
| Stop Remote Node | 0 | 02h | Node-ID* | Sets the CANopen communication state machine on the designated node to Stopped. |
| Pre-Operational State | 0 | 80h | Node-ID* | Sets the CANopen communication state machine on the designated node to Pre-Operational. In the pre-operational state, only NMT and SDO messages are allowed. |
| Reset Node | 0 | 81h | Node-ID* | Resets the designated node (same as power cycle). Results in a Boot Up message sent by the node. |
| Reset Communication | 0 | 82h | Node-ID* | Resets CANopen communication state machine on the designated node. Results in a Boot Up message sent by the node. |

*Node-ID = Drive address (1…7Fh)

### *4.1.1 Boot Up State*

Upon power-up, each drive initializes by going through the Reset Node and Reset Communication states. If the initialization process succeeds, the drive sends out a Boot-Up message and goes into the Pre-Operational state.

### *4.1.2 Pre-Operational State*

Communication is limited to all message types except PDO messages. In this state, the NMT master can command the communication state machine to enter any of the states listed in Table 3 above. Generally, the host keeps a node in pre-operational state during setup and configuration.

### *4.1.3 Operational State*

Enables all message types including PDO messages. In this state, the NMT master can command the communication state machine to enter any of the states listed in Table 2.

### *4.1.4 Stopped State*

Disables all message types except NMT messages; Node Guarding / Life Guarding (see below) remains active.

### *4.1.5 NMT Message Examples*

| COB-ID | Number of Bytes | Message / Data | Description |
|---|---|---|---|
| 000 | 2 | 80 01 | Host: NMT Host commands node 1 into Pre-Operational state |
| 000 | 2 | 01 01 | Host: NMT Host commands node 1 into Operational state |
| 000 | 2 | 02 01 | Host: NMT Host commands node 1 into Pre-Operational state |
| 000 | 2 | 82 01 | Host: NMT Host commands a Reset to Node 1 |
| 701 | 1 | 00 | Node 1 response: Cycles through the standard boot-up states stopping in the Pre-operational state. The control state machine is also reset. This is the same as a power cycle |
| 000 | 2 | 81 01 | Host: NMT Host commands Communication Reset |

| | | | |
|---|---|---|---|
| 701 | 1 | 00 | Node 1 response: Cycles through the standard boot-up states stopping in the Pre-operational state. The control state machine does not reset and retains full motion control. |

## 4.2 NMT Error Control

AMC CANopen drives support Node Guarding and Life Guarding, as NMT error controls.

### *4.2.1 Node Guarding*

The NMT Master can monitor the communication status of each node using the Node Guarding protocol. During node guarding, a drive is polled periodically and is expected to respond with its communication state within a pre-defined time frame. Acceptable states are shown in Table 3. Note that responses indicating an acceptable state will alternate between two different values due to a toggle bit in the returned value. If there is no response, or an unacceptable state occurs, the NMT master reports an error to its host application. The Node Guard message is sent at time intervals, determined by the Guard Time (object 100Ch). The NMT slave (node) must reply to this message before the end of this time interval. Figure 6 and Figure 7 show the message format for an NMT master request and the correct NMT slave response. Note that the slave always responds with a toggle bit in byte 1, therefore the response will toggle between the two values shown in Table 3.

### *4.2.2 Life Guarding*

Similarly, the NMT slave monitors the status of the NMT master (Life Guarding). This event utilizes the Guard Time (object 100Ch) and Life Time Factor (object 100Dh) to determine a “Lifetime” for each NMT slave (Lifetime = Guard Time X Life Time Factor). If a node does not receive a Node Guard message within its Lifetime, the node assumes communication with the host is lost and performs an error event (set by object 2065h). Each node may have a different Lifetime.

**Figure 6: NMT master Node Guard request (host to node).**

| Arbitration Field | | Data Field | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| COB-ID | RTR | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
| 700h + Node-ID | 1 | These bytes not sent | | | | | | | |

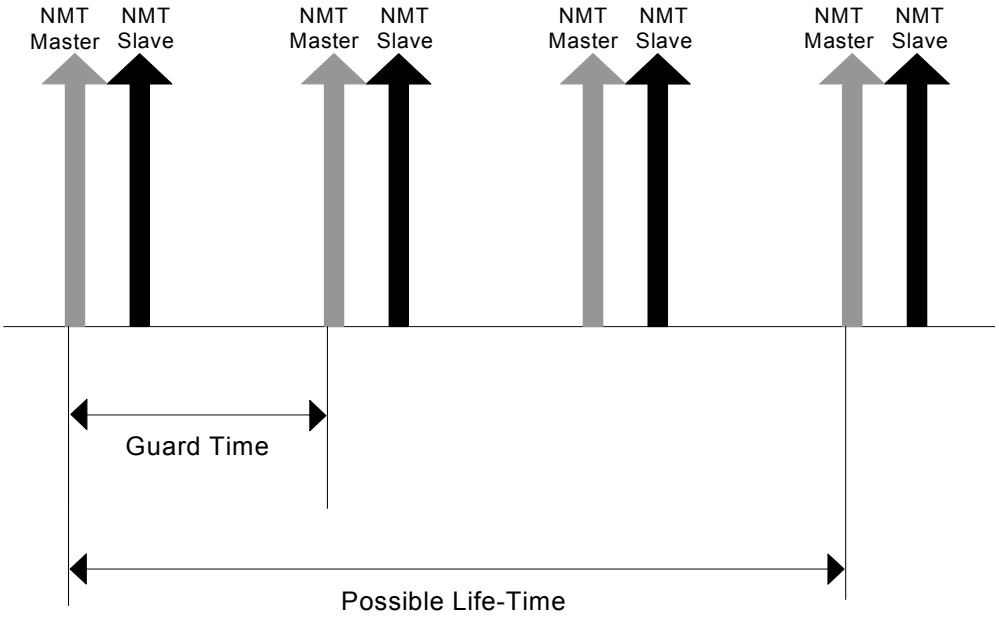
**Figure 7: NMT slave Node Guard reply (node to host).**

| Arbitration Field | Data Field | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| COB-ID | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
| 700h + Node-ID | See Table 3 | These bytes not sent | | | | | | |

**Table 3: Acceptable NMT slave return values.**

| Return Value | Communication Status |
|---|---|
| 4h or 84h | STOPPED |
| 5h or 85h | OPERATIONAL |
| 7Fh or FFh | PRE-OPERATIONAL |

**Figure 8: Example of Guard Time and Life Time. The first grey arrow represents an NMT request from the master and the second black arrow represents an NMT response from the slave. In this case, the Life Time is a factor of 3X greater than the Guard Time.**




### *4.2.3 Node Guard / Lifegaurd Example*

In this example, NMT messages are used to transition the Communication states of the drive while NodeGuarding is active. The shaded rows indicate how the node will respond to a given host command.

| COB-ID | Number of Bytes | Message / Data | Description |
|---|---|---|---|
| 701 | 0 | RTR set | Host sends first node guard message within GuardTime |
| 701 | 1 | 04 | Node replies in STOP state |
| 701 | 0 | RTR set | Host sends next node guard message within GuardTime |
| 701 | 1 | 84 | Node replies in STOP state, Toggle Bit alternates |
| 701 | 0 | RTR set | Host sends next node guard message within GuardTime |
| 701 | 1 | 04 | Node replies in STOP state, Toggle Bit alternates |
| 000 | 2 | 80 01 | NMT host changes node communication state machine to Pre-Operational |
| 701 | 0 | RTR set | Host sends next node guard message within GuardTime |
| 701 | 1 | FF | Node replies in PRE-Operational state, Toggle Bit alternates |
| 701 | 0 | RTR set | Host sends next node guard message within GuardTime |
| 701 | 1 | 7F | Node replies in PRE-Operational state, Toggle Bit alternates |
| 000 | 2 | 01 01 | NMT host changes node communication state machine to Operational |
| 701 | 1 | RTR set | Host sends next node guard message within GuardTime |
| 701 | 0 | 85 | Node replies in Operational state, Toggle Bit alternates |
| 701 | 1 | RTR set | Host sends next node guard message within GuardTime |

| | | | |
|---|---|---|---|
| 701 | 0 | 05 | Node replies in Operational state, Toggle Bit alternates |

## 4.3 BOOT-UP Message

The drive transmits a boot-up message after power up, communication reset, or application reset events. The CANopen master can monitor the drive and report an error if no boot-up message was received. The boot-up message of an AMC CANopen drive uses the same COB-ID as a Node Guard reply.

**Table 4: Boot-up message from AMC CANopen drives.**

| Arbitration Field | Data Field | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| COB-ID | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
| 700h + Node-ID | 00 | These bytes not sent | | | | | | |

### *4.3.1 Boot-Up Example*

These are messages sent from three drives powered up in random order. Data is always 00h for boot up messages.

| COB-ID | Number of Bytes | Message / Data | Description |
|---|---|---|---|
| 701 | 1 | 00 | Node 1 boots up |
| 703 | 1 | 00 | Node 3 boots up |
| 702 | 1 | 00 | Node 2 boots up |

## 4.4 SYNC Message

The SYNC message serves as a network “trigger” and is used to coordinate events across multiple CANopen nodes. For example, the CANopen host may need to obtain the actual motor position at a specific time, for several nodes. An AMC CANopen drive can be pre-configured to read and broadcast its actual position the instant a SYNC message is received. SYNC messages carry no data. AMC drives receive SYNC messages, but cannot produce them. For more information on the SYNC message, see (DS301).

**Table 5: Sync message format (host to node).**

| Arbitration Field | | Data Field | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| COB-ID | RTR | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
| 80h | 0 | These bytes not sent | | | | | | | |

### *4.4.1 SYNC Message Example*

In this example TPDO1 (1800.02h) is configured to report the StatusWord every second Sync message the host broadcasts. This example starts with the host setting Node 1 into the Operational state so PDOs may be processed by the drive.

| COB-ID | Number of Bytes | Message / Data | Description |
|---|---|---|---|
| 000 | 2 | 01 01 | Host: NMT command puts Node 1 into Operational state. |
| 80 | 0 | None | Host: 1st Sync message |
| 80 | 0 | None | Host: 2nd Sync message |

| | | | |
|---|---|---|---|
| 231 | 2 | 60 06 | Node 1 response: TPDO1 (1A00.01h) sends data containing StatusWord |
| 80 | 0 | None | Host: 3rd Sync message |
| 80 | 0 | None | Host: 4th Sync message |
| 231 | 2 | 60 06 | Node 1 response: TPDO1 (1A00.01h) sends data containing StatusWord |

## 4.5 EMERGENCY Messages

EMERGENCY messages are sent by the CANopen nodes to provide important status information to the CANopen host controller. An emergency object is transmitted only once per error event by the drive, and uses the same COB-ID as the sync message plus the node ID. AMC servo drives utilize EMERGENCY messages to indicate PVT buffer status information to the CANopen host controller. The following tables describe the error codes supported by AMC CANopen drives.

**Table 6 Emergency Object Data**

| Arbitration Field | Data Field | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| COB-ID | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
| 80h + Node-ID | 00 | 00 | 00 | Error Code. See (Table 7). | See (Table 7) | | | |

### *4.5.1 EMERGENCY Error Codes*

**Table 7: Emergency Error Codes supported by AMC CANopen drives.**

<table>
<tr><th>Error Code</th><th>Description</th><th>Bytes 5 – 8</th></tr>
<tr><td>00h</td><td>PVT Sequence Counter Error</td><td>Required counter value</td></tr>
<tr><td>01h</td><td>PVT Cannot be started</td><td>Internal use only</td></tr>
<tr><td>02h</td><td>PVT Buffer Underflow</td><td>0h</td></tr>
<tr><td>80h - FFh</td><td>RPDO Cannot be Processed<br><br>Bit Definitions are defined as follows when Bit 7 = 1<br><br>Bits 4 - 6 = Subtract 1 from the value read in these bits to get the Sub-index of the RPDO Mapping Parameter that caused the error.<br>Bits 0 - 3 = Error Description Values (1h - 7h) where:
<table>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>General Error</td></tr>
<tr><td>2</td><td>Object does not exist</td></tr>
<tr><td>3</td><td>Not writable or Not readable</td></tr>
<tr><td>4</td><td>Access unsupported in present state</td></tr>
<tr><td>5</td><td>Not enough space in the PDO for object data</td></tr>
<tr><td>6</td><td>Data integrity error</td></tr>
<tr><td>7</td><td>Internal write error</td></tr>
</table>
</td><td>COB-ID of RPDO</td></tr>
</table>

### *4.5.2 EMERGENCY Message Examples*

These examples demonstrate several emergency messages and what the data will look like coming from the drive.

| COB-ID | Number of Bytes | Message / Data | Description |
|---|---|---|---|
| 81 | 8 | 00 00 00 00 03 00 00 00 | The 3rd counter value was skipped when filling the PVT buffer of Node 1. |
| 83 | 8 | 00 00 00 01 00 00 00 00 | PVT cannot be started on node 3. It happens to be in the wrong state here. |
| 81 | 8 | 00 00 00 84 01 05 00 00 | 84 indicates an RPDO that cannot be processed because access is not supported in the present state. 0501 indicates the COB-ID of the RPDO. This message occurred because write access to the drive was disabled before attempting to write. |

## 4.6 TIME STAMP Message

The TIME STAMP message provides a “global clock” for all the nodes on the CANopen network. The TIME STAMP message data field contains the host controller time. It is used for synchronization between nodes. This can be very important for applications that require long-term time synchronization.

Each drive uses not only the time data contained in the timestamp messages, but also the time between each timestamp message to synchronize to both host timing and frequency. If there is jitter in the host’s timestamp messages, there will be some jitter in the drive timing.

The data field uses a 6 byte “Time Of Day” field defined in CiA’s DS301. Time Of Day contains two components: the number of milliseconds after midnight (4 bytes), and the present day since January 1, 1984.

**Figure 9: Timestamp message data.**

| Arbitration Field | | Data Field | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| COB-ID | RTR | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
| 100h | 0 | Time, after Midnight in Milliseconds (LSB first) | | | | Current day since 01/01/84 | | N/A | N/A |

### *4.6.1 TIME STAMP Example*

This example starts the drive at midnight on the 1st day of January 1984 as dictated by the CiA’sDS301. Generally the current time and day would be filled in and sent automatically. AMC CANopen servo drives do not respond to timestamps with messages, therefore there is no node response shown.

| COB-ID | Number of Bytes | Message / Data | Description |
|---|---|---|---|
| 100 | 8 | 00 00 00 00 00 00 00 00 | Very first timestamp Resets timers on all nodes to the value contained in bytes 1 – 6 |
| Wait 500 ms | | | |
| 100 | 8 | F4 01 00 00 00 00 00 00 | Broadcast message reporting time is now 500 ms later |
| Wait 500 ms | | | |
| 100 | 8 | DC 05 00 00 00 00 00 00 | Broadcast message reporting time is now 500 ms later |
| Wait 500 ms | | | |
| 100 | 8 | D0 07 00 00 00 00 00 00 | Broadcast message reporting time is now 500 ms later |
| Wait 500 ms | | | |
| 100 | 8 | C4 09 00 00 00 00 00 00 | Broadcast message reporting time is now 500 ms later |

| Wait 500 ms | | | |
|---|---|---|---|
| 100 | 8 | B8 0B 00 00 00 00 00 00 | Broadcast message reporting time is now 500 ms later |
| Wait 500 ms | | | |
| 100 | 8 | AC 0D 00 00 00 00 00 00 | Broadcast message reporting time is now 500 ms later |

## 4.7 SDO vs. PDO Messages

There are two methods for reading and writing data to objects: Service Data Object (SDO) and Process Data Object (PDO) messages. An SDO consists of an outgoing message from host to node, possibly some intermediate messages between host and node, and a reply message from node to host; this is referred to as confirmed messaging. A PDO consists of a single unconfirmed message that requires less bus traffic relative to its SDO counterpart. Although PDOs make more efficient use of the CAN bus than do SDOs, PDO messages must be configured prior to using (see PDO Configuration). Furthermore, PDOs are restricted to the transmission of no more than 8 bytes whereas there is no limitation to the number of bytes SDOs can transfer. SDO messages may be used any time but are generally used before actual drive operation for set-up and configuration. PDO messages are generally used during drive operation, such as for setting target commands.

## 4.8 SDO Messages

AMC CANopen servo drives support read and write SDO messages that can be divided into 4 categories:

- Reading objects that contain 4 or less data bytes (expedited read)
- Writing to objects that contain 4 or less data bytes (expedited write)
- Reading objects that contain more than 4 data bytes (segmented read)
- Writing to objects that contain more than 4 data bytes (segmented write)

The first data byte in the Data field, called the 'command' byte, is used to determine any of the above possible cases. Then, depending upon the particular case, the next 3 bytes may be used to specify an object index with 4 bytes left for object data or all 7 remaining bytes may be used purely for object data. It is important to distinguish between the data bytes of the Data field and the data bytes of an object. The data bytes of the Data field are the 8 bytes of a CAN frame whereas the object data bytes refer to the information stored in an object.

Of the bytes used for object data, only some may be used with the others left empty (equal to zero). For example, if an SDO message is used to read an object with only 2 bytes of information, then only two of the data bytes in the returned message will contain the relevant data while the others will be left equal to zero. However, there may be cases where the relevant data is also equal to zero. In this case, there must be a way to distinguish relevant data bytes from empty data bytes. If the message recipient knows how many bytes to expect, then there is no issue. Otherwise, size indication is needed. Although size indication is specified in DS301 it is also not required. To comply with this, AMC CANopen drives offer an SDO Size Indication object (2111h) for enabling and disabling size indication as defined by DS301.

### *4.8.1 Expedited SDO Messages*

This is a 1-step process and applies only when reading / writing objects with 4 or less data bytes (e.g. 8-bit, 16-bit, 32-bit data types). Expedited messages are simple read / write commands where the complete set of data is included in the last four bytes of the message (write command), or the last 4 bytes of the reply (read command). Whether the host is reading or writing to a node, the process requires only one command and one reply.

### *4.8.2 Segmented SDO Messages*

This is a multi-step process that applies when reading / writing messages larger than 4 bytes (e.g. string).

Step 1, called "initiation," is merely handshaking between the host and node. To initialize communication, the host gives a command, and the node responds confirming that it is ready for data exchange. No data is exchanged during the initiation step.

The next steps are the actual data exchange. This can include many messages between the host and the node. The command byte, in these steps, contains a “Toggle Bit” and “Last Segment” bit. In these steps, every message the host sends to the drive must alternate the toggle bit (this is done automatically by following the procedures for message construction below). The last segment bit is only set to 1 when the current message contains the last of the data to transfer; this indicates that the process is finished.

**Figure 10: Expedited SDO Read (4 or less data bytes)**

| SDO READ, EXPEDITED (4 or less bytes) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Step 1a: Host initiates Read command** | | | | | | | | |
| **Arbitration Field** | **Data Field** | | | | | | | |
| **COB-ID** | **Byte 1** | **Byte 2** | **Byte 3** | **Byte 4** | **Byte 5** | **Byte 6** | **Byte 7** | **Byte 8** |
| 600h + Node-ID* | 40h | Object Index (LSB) | Object Index (MSB) | Sub-Index | Use 00h for all 4 bytes | | | |
| **Step 1b: Node Replies to host with all data** | | | | | | | | |
| **Arbitration Field** | **Data Field** | | | | | | | |
| **COB-ID** | **Byte 1** | **Byte 2** | **Byte 3** | **Byte 4** | **Byte 5** | **Byte 6** | **Byte 7** | **Byte 8** |
| 580h + Node-ID* | 42h, 4Fh, 4Bh, or 43h See Table 8 | Object Index (LSB) | Object Index (MSB) | Sub-Index | Data, LSB first | | | |

*Node-ID is node address (0…7Fh)

**Figure 11: Host to node Initiate read, more than 4 bytes**

| SDO READ, SEGMENTED (more than 4 bytes) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **STEP 1a. Host request for data** | | | | | | | | |
| **Arbitration Field** | **Data Field** | | | | | | | |
| **COB-ID** | **Byte 1** | **Byte 2** | **Byte 3** | **Byte 4** | **Byte 5** | **Byte 6** | **Byte 7** | **Byte 8** |
| 600h + Node-ID* | 40h | Object Index (LSB) | Object Index (MSB) | Sub-Index | Use 00h for all 4 bytes | | | |
| **STEP 1b. Node reply, ready to transmit data** | | | | | | | | |
| **Arbitration Field** | **Data Field** | | | | | | | |
| **COB-ID** | **Byte 1** | **Byte 2** | **Byte 3** | **Byte 4** | **Byte 5** | **Byte 6** | **Byte 7** | **Byte 8** |
| 580h + Node-ID* | 40h or 41h See Table 8 STEP 1 | Object Index (LSB) | Object Index (MSB) | Sub-Index | 00h or Number of bytes to transfer | | | |
| **STEP 2a. Host confirms, ready for data** | | | | | | | | |
| **Arbitration Field** | **Data Field** | | | | | | | |
| **COB-ID** | **Byte 1** | **Byte 2** | **Byte 3** | **Byte 4** | **Byte 5** | **Byte 6** | **Byte 7** | **Byte 8** |
| 600h + Node-ID* | 60h See Table 8 STEP 2 | Use 00h for all 7 bytes | | | | | | |

| STEP 2b. Node replies with data | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Arbitration Field** | **Data Field** | | | | | | | |
| **COB-ID** | **Byte 1** | **Byte 2** | **Byte 3** | **Byte 4** | **Byte 5** | **Byte 6** | **Byte 7** | **Byte 8** |
| 580h + Node-ID* | See Table 8 STEP 2 | Data, LSB first | | | | | | |

*Node-ID is node address (0…7Fh)

**Table 8: READ Command (Byte 1) values and their meaning**

| Usage | Command Byte values | Meaning |
|---|---|---|
| Read SDO **Step 1** | 40h | Always used by host when initiating read process. Does not include size indication. Used by node when replying to hosts' initiate read command, but only when object 2111h = 0 and there are more than 4 bytes to transfer. |
| | 41h | Used by node only when replying to read initiation and there are more than 4 bytes to transfer. Bytes 5 – 8 will indicate number of bytes the node has to transfer (LSB first). Only occurs if object 2111h ≠ 0, otherwise node will reply with 40h instead. |
| | 42h | Used by node when replying to read command with 4 or less data bytes in 5 – 8 (LSB first). Actual number of valid bytes is not indicated. Only occurs if object 2111h = 0. |
| | 4Fh | Used by node when replying to read command with exactly 1 data byte, i.e. reading an 8-bit object. Use only byte 5 (ignore 6 - 8). Only occurs if object 2111h ≠ 0, otherwise node will use 42h. |
| | 4Bh | Used by node when replying to read command with exactly 2 data bytes in bytes 5 and 6, i.e. reading a 16-bit object (ignore 7 and 8). Only occurs if object 2111h ≠ 0, otherwise node will use 42h. |
| | 43h | Used by node when replying to read command with exactly 4 data bytes in bytes 5 – 8, i.e. reading a 32-bit object. Only occurs if object 2111h ≠ 0, otherwise node will use 42h. |
| Read SDO **Step 2** Only data transfers larger than 4 bytes | 60h | Used by host. Second step to "Segmented" read process always begins with 60h. Each time the node replies with data, the host must toggle between 60h and 70h. If the host does not toggle between two consecutive messages, the node will abort transfer with 80h. |
| | 70h | |
| | 0h | Reply from node. Will only occur if host used 60h in the previous command and there is more data to transmit. In this case the host should send another message using 70h in byte 1 and 00h for all other bytes to retrieve more data. |
| | 1h | Reply from node. Will only occur if host used 60h in the previous command and this message contains the last of the data. |
| | 10h | Reply from node. Will only occur if host used 70h in the previous command and there is more data to transmit. In this case the host should send another message using 60h in byte 1 and 00h for all other bytes to retrieve more data. |
| | 11h | Reply from node. Will only occur if host used 70h in the previous command and this message contains the last of the data. |
| | 3h, 5h, 7h, 9h, Bh, Dh | Same as 1h except the number of bytes not containing data is specified. 3h if only the last byte contains no data, 5h if only the last two bytes do not contain data, and onwards up to Dh if the last 6 bytes do not contain data. Only occurs if object 2111h ≠ 0, otherwise node will reply with 1h. |
| | 13h, 15h, 17h, 19h, 1Bh, 1Dh | Same as 11h except the number of bytes not containing data is specified. 13h if only the last byte contains no data, 15h if only the last two bytes do not contain data, and onwards up to 1Dh if the last 6 bytes do not contain data. Only occurs if object 2111h ≠ 0, otherwise node will reply with 11h. |

**Figure 12: Expedited SDO Write (4 or less data bytes)**

| SDO WRITE, EXPEDITED (4 or less data bytes) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Step 1a: Host initiates write command with data** | | | | | | | | |
| **Arbitration Field** | **Data Field** | | | | | | | |
| **COB-ID** | **Byte 1** | **Byte 2** | **Byte 3** | **Byte 4** | **Byte 5** | **Byte 6** | **Byte 7** | **Byte 8** |
| 600h + Node-ID* | 22h, 2Fh, 2Bh, or 23h See Table 9 | Object Index (LSB) | Object Index (MSB) | Sub-Index | Data, LSB first | | | |
| **Step 1b: Node Replies to host with all data** | | | | | | | | |
| **Arbitration Field** | **Data Field** | | | | | | | |
| **COB-ID** | **Byte 1** | **Byte 2** | **Byte 3** | **Byte 4** | **Byte 5** | **Byte 6** | **Byte 7** | **Byte 8** |
| 580h + Node-ID* | 60h See Table 9 | Object Index (LSB) | Object Index (MSB) | Sub-Index | Ignore | | | |

*Node-ID is node address (0…7Fh)

**Figure 13: Host to node Initiate write, more than 4 bytes**

| SDO WRITE, SEGMENTED (more than 4 data bytes) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **STEP 1a. Host initiates data transfer** | | | | | | | | |
| **Arbitration Field** | **Data Field** | | | | | | | |
| **COB-ID** | **Byte 1** | **Byte 2** | **Byte 3** | **Byte 4** | **Byte 5** | **Byte 6** | **Byte 7** | **Byte 8** |
| 600h + Node-ID* | 20h or 21h See Table 9 | Object Index (LSB) | Object Index (MSB) | Sub-Index | 00h or Number of bytes to transfer | | | |
| **STEP 1b. Node reply, ready to accept data** | | | | | | | | |
| **Arbitration Field** | **Data Field** | | | | | | | |
| **COB-ID** | **Byte 1** | **Byte 2** | **Byte 3** | **Byte 4** | **Byte 5** | **Byte 6** | **Byte 7** | **Byte 8** |
| 580h + Node-ID* | 60h See Table 9 | Object Index (LSB) | Object Index (MSB) | Sub-Index | 00h | | | |
| **STEP 2a. Host begins data transfer** | | | | | | | | |
| **Arbitration Field** | **Data Field** | | | | | | | |
| **COB-ID** | **Byte 1** | **Byte 2** | **Byte 3** | **Byte 4** | **Byte 5** | **Byte 6** | **Byte 7** | **Byte 8** |
| 600h + Node-ID* | 0h, 1h, 10h, 11h See Table 9 | Data, LSB first | | | | | | |

| STEP 2b. Node replies | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Arbitration Field** | **Data Field** | | | | | | | |
| **COB-ID** | **Byte 1** | **Byte 2** | **Byte 3** | **Byte 4** | **Byte 5** | **Byte 6** | **Byte 7** | **Byte 8** |
| 580h + Node-ID* | 20h, or 30h See Table 9 | Ignore | | | | | | |

**Table 9: WRITE Command (Byte 1) values and their meaning**

| Usage | Command Byte values | Meaning |
|---|---|---|
| Host Initiates Write SDO more than 4 data bytes | 20h | Used by host when initiating a write process of more than 4 data bytes. Total number of bytes is not indicated. Node replies with 60h, confirming that it is ready to receive data. |
| | 21h | Used by host when initiating a write process of more than 4 data bytes. Total number of bytes is indicated using bytes 5 – 8 (LSB first). Node replies with 60h, confirming that it is ready to receive data. Only use if object 2111h ≠ 0, otherwise use 20h. |
| Host Initiates Write SDO 4 or less data bytes | 22h | Used by host when writing 4 or less data bytes. Total number of data bytes not indicated. Node replies with confirmation 60h. |
| | 2Fh | Used by host when writing exactly 1 data byte. Byte 5 contains data. Node replies with confirmation 60h. Only use if object 2111h ≠ 0, otherwise use 22h. |
| | 2Bh | Used by host when writing exactly 2 data bytes. Byte 5 and 6 contain data. Node replies with confirmation 60h. Only use if object 2111h ≠ 0, otherwise use 22h. |
| | 23h | Used by host when writing exactly 4 data bytes. Bytes 5 – 8 contain data. Node replies with confirmation 60h. Only use if object 2111h ≠ 0, otherwise use 22h. |
| Data transfer commands | 60h | Reply from node. 60h only occurs once during the initiate write process, after that each consecutive reply to a message containing data will toggle between 20h and 30h. 20h always occurs first after 60h. |
| | 20h | |
| | 30h | |
| | 00h | Used by host if the nodes previous reply contained 60h or 30h in byte 1 and there is still data left to transmit. |
| | 1h | Used by host if the nodes previous reply contained 60h or 30h in byte 1 and this message contains the last data to transfer. |
| | 10h | Used by host if the nodes previous reply contained 20h in byte 1 and there is still data left to transmit. |
| | 11h | Used by host if the nodes previous reply contained 20h in byte 1 and this message contains the last data to transfer. |
| | 3h, 5h, 7h, 9h, Bh, Dh | Same as 1h except the number of bytes not containing data is specified. 3h if only the last byte contains no data, 5h if only the last two bytes do not contain data, and onwards up to Dh if the last 6 bytes do not contain data. Only use if object 2111h ≠ 0, otherwise use 1h. |
| | 13h, 15h, 17h, 19h, 1Bh, 1Dh | Same as 11h except the number of bytes not containing data is specified. 13h if only the last byte contains no data, 15h if only the last two bytes do not contain data, and onwards up to 1Dh if the last 6 bytes do not contain data. Only use if object 2111h ≠ 0, otherwise use 11h. |

### *4.8.3 SDO Abort Transfer Messages*

When an error occurs during reading or writing an object, the node sends an abort transfer message to the host.

**Figure 14: Node indicates error in communication.**

| Arbitration Field | Data Field | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| COB-ID | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
| 80h + Node-ID | 80h | Object Index (LSB) | Object Index (MSB) | Sub-Index | See Abort Code Table below (LSB first) | | | |

**Table 10: Abort Code descriptions**

| Abort Code | Description |
|---|---|
| 0503 0000h | Toggle bit not alternated |
| 0504 0000h | SDO protocol timed out |
| 0504 0001h | Command specifier not valid |
| 0504 0002h | Invalid block size (block mode only, see DS301) |
| 0504 0003h | Invalid sequence number (block mode only, see DS301) |
| 0504 0004h | CRC error (block mode only, see DS301) |
| 0504 0005h | Out of memory |
| 0601 0000h | Unsupported access to an object |
| 0601 0001h | Attempt to read a write only object |
| 0601 0002h | Attempt to write a read only object |
| 0602 0000h | Object does not exist in the object dictionary |
| 0604 0041h | Object cannot be mapped to the PDO |
| 0604 0042h | The number and length of the objects to be mapped would exceed PDO length |
| 0604 0043h | General parameter incompatibility reason |
| 0604 0047h | General internal incompatibility in the device |
| 0606 0000h | Access failed due to a hardware error |
| 0607 0010h | Data type does not match, length of service parameter does not match |
| 0607 0012h | Data type does not match, length of service parameter too high |
| 0607 0013h | Data type does not match, length of service parameter too low |
| 0609 0011h | Sub-index does not exist |
| 0609 0030h | Value range of parameter exceeded (only for write access) |
| 0609 0031h | Value of parameter written too high |
| 0609 0032h | Value of parameter written too low |
| 0609 0036h | Maximum value is less than minimum value |
| 0800 0000h | General error |
| 0800 0020h | Data cannot be transferred or stored to the application |
| 0800 0021h | Data cannot be transferred or stored to the application because of local control |
| 0800 0022h | Data cannot be transferred or stored to the application because of present device state |
| 0800 0023h | Object dictionary dynamic generation fails or no object dictionary is present (object dictionary loads from file and file error occurred) |

### *4.8.4 SDO Read and Write Examples*

#### 4.8.4.1 EXPEDITED SDO READ EXAMPLE

In this example, Size indication (object 2111h) is turned **off** so that the drive **will not indicate**, in any message, how many valid bytes are contained in the message. In this case the user is responsible for knowing the message size.

| COB-ID | Number of Bytes | Message / Data | Description |
|---|---|---|---|
| 601 | 8 | 40 64 60 00 00 00 00 00 | Host uses 40 in the command byte (see Table 8) to read object 6064h, the 3rd data byte is zero because this object has no sub-indices and the last 4 data bytes are don't care's when reading |
| 581 | 8 | 42 64 60 00 34 33 00 00 | Node replies with 42 because size indication is off (see Table 8) and message was received as an expedited data transfer. Bytes 5 – 8 will contain the data from the object. In this case object 6064h (Actual Position) contains 00 00 33 34h (13,108 in decimal). |

#### 4.8.4.2 EXPEDITED SDO WRITE EXAMPLE

In this example, Size indication (object 2111h) is turned **off** so that the drive **will not indicate**, in any message, how many valid bytes are contained in the message. When writing data to a node, it is not required for the host to use size indications in the messages to the node. In this case the user is responsible for knowing the message size and for using the command byte 22h.

| COB-ID | Number of Bytes | Message / Data | Description |
|---|---|---|---|
| 601 | 8 | 22 40 60 00 0F 00 00 00 | Host uses 22 in the command byte (see Table 9) to write object 6040h, the 3rd data byte is zero because this object has no sub-indices. The last 4 data bytes contain the data to write to the object. |
| 581 | 8 | 60 40 60 00 00 00 00 00 | Node replies with 60 (see Table 9) indicating message was received. Bytes 1-3 contain the object index and sub-index. Bytes 4 – 7 will always be zero in this case |

#### 4.8.4.3 SEGMENTED SDO READ EXAMPLE

In this example, Size indication (object 2111h) is turned **on** so that the drive **will indicate**, in any message that contains less than 7 data bytes, how many valid bytes are contained in the message. Node replies to each host message are shaded. When the data from the last 5 shaded rows is concatenated and converted to ASCII according to object 208D.01, this data reads "Ccr300ee 3.0.0."

| COB-ID | Number of Bytes | Message / Data | Description |
|---|---|---|---|
| 601 | 8 | 40 8D 20 01 00 00 00 00 | Host begins data transfer Initialization |
| 581 | 8 | 41 8D 20 01 20 00 00 00 | Node replies with 41 indicating there are more than 4 bytes to transfer. Bytes 4 – 7 indicate the number of bytes necessary to transfer. In this case 20h = 32 bytes. The drive now waits for the host to begin data transfer confirmation. |
| 601 | 8 | 60 00 00 00 00 00 00 00 | Host uses 60 to confirm ready for first segment. All other bytes are zero |
| 581 | 8 | 00 43 63 72 33 30 30 65 | Node responds to host with 00h and 7 data bytes. |
| 601 | 8 | 70 00 00 00 00 00 00 00 | Host uses 70 to confirm ready for next segment. All other bytes are zero |
| 581 | 8 | 10 65 20 33 2E 30 2E 30 | Node responds to host with 10h and 7 data bytes. |

| | | | |
|---|---|---|---|
| 601 | 8 | 60 00 00 00 00 00 00 00 | Host uses 60 to confirm ready for next segment. All other bytes are zero |
| 581 | 8 | 00 00 00 00 00 00 00 00 | Node responds to host with 00h and 7 data bytes. |
| 601 | 8 | 70 00 00 00 00 00 00 00 | Host uses 70 to confirm ready for next segment. All other bytes are zero |
| 581 | 8 | 10 00 00 00 00 00 00 00 | Node responds to host with 10h and 7 data bytes. |
| 601 | 8 | 60 00 00 00 00 00 00 00 | Host uses 60 to confirm ready for next segment. All other bytes are zero |
| 581 | 8 | 07 00 00 00 00 00 00 00 | Node responds to host with 07h and 7 data bytes. The 07h indicates that the last three bytes are to be ignored. |

#### 4.8.4.4 SEGMENTED SDO WRITE EXAMPLE

In this example, Size indication (object 2111h) is turned **on** so that the drive **will indicate**, in any message that contains less than 7 data bytes, how many valid bytes are contained in the message. When writing data to a node, it is not required for the host to use size indications in the messages to the node. Node replies to each host message are shaded. Data must be sent to the node according to each objects required format. See the Object dictionary for more information on writing to a specific object.

| COB-ID | Number of Bytes | Message / Data | Description |
|---|---|---|---|
| 601 | 8 | 20 0B 20 01 00 00 00 00 | Host begins data transfer Initialization |
| 581 | 8 | 60 0B 20 01 00 00 00 00 | Node replies with 60 confirming message receipt and ready for first segment. |
| 601 | 8 | 00 57 69 6C 6C 20 45 6C | Host uses 00 to begin data transfer protocol. Last 7 bytes contain data. |
| 581 | 8 | 20 57 69 6C 00 00 00 00 | Node responds to host with 20h. Ignore Last 7 bytes |
| 601 | 8 | 11 6B 69 6E 73 20 45 6C | Host uses 11 to indicate “Last Segment”. Any bytes that are more than an objects length will no be written. |

## 4.9 PDO Messages

PDO messages exchange information between the host and nodes without the overhead of SDO messages. PDO messages have no reply, (i.e. they are unconfirmed messages) which allows for fast, efficient data transfer of up to 8 bytes. As a result, PDOs are ideal for transferring information during device operation whereas SDOs are generally used for configuring the drive. PDO messages, unlike SDO messages, are configured prior to use. Once configured, PDO messages can be enabled or disabled according to whether or not they are needed. There are two types of PDO messages: a transmit PDO (TPDO) message and a receive PDO (RPDO) message.

### *4.9.1 Transmit Process Data Objects (TPDO)*

TPDOs are configured to send data from node to host according to a configurable trigger mechanism or when requested by an RTR. Before data is transmitted by a TPDO, it must be configured, and enabled, with the The Communication Parameter Object related to that TPDO. TPDOs do not alter any object data; they only read and transmit data to the CANbus. AMC CANopen drives offer ten different TPDOs (all are disabled by default). Nine have fixed pre-defined configurations and one is available for user specification.

### *4.9.2 Receive Process Data Objects (RPDO)*

The host uses RPDOs to write data to objects in one or more nodes. Before data is received by an RPDO, it must be configured, and enabled, with a The Communication Parameter Object related to that RPDO. Since RPDOs write to object data, it is important to ensure that the data sent is in agreement with the objects

mapped to the PDO (PDO object mapping is discussed below). AMC CANopen drives offer nine different RPDOs where all are disabled by default.

### 4.9.3 PDO Configuration

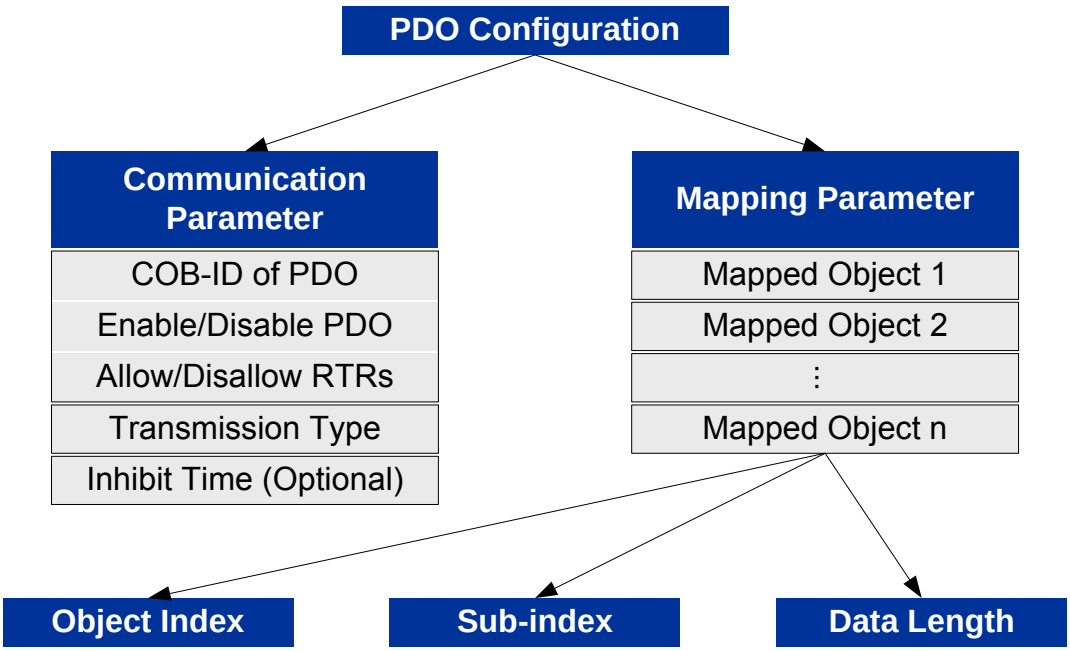
Configuration of a particular PDO is accomplished by setting the appropriate PDO The Communication Parameter Object and PDO Mapping Parameter object The Mapping Parameter Object for that PDO. It is the users responsibility to decide which of the PDOs in Table 11 are applicable to the application and configure/enable them. As specified by DS301, the PDO Communication Parameter objects are found over the range 1400h-15FFh and 1800h-19FFh for RPDOs and TPDOs, respectively. PDO Mapping Parameter objects are specified over the range 1600h-17FFh and 1A00h-1BFFh for RPDOs and TPDOs, respectively. Although the full range allows for over 500 different RPDOs and TPDOs, only a fraction of that range is needed for AMC CANopen drives. The PDOs used by AMC CANopen drives are given in Table 11 along with the names of objects mapped to them. Only one TPDO (26th) is user map-able; all other TPDOs and RPDOs have fixed mapping parameters.

**Table 11**

| PDO | PDO Communication Parameter | PDO Mapping Parameter | 1st Object Mapping | 2nd Object Mapping |
|---|---|---|---|---|
| 1st RPDO | 1400h | 1600h | ControlWord | - |
| 2nd RPDO | 1401h | 1601h | ControlWord | Modes of Operation |
| 3rd RPDO | 1402h | 1602h | ControlWord | Target Position |
| 4th RPDO | 1403h | 1603h | ControlWord | Target Velocity |
| 5th RPDO | 1404h | 1604h | ControlWord | Target Current |
| 21st RPDO | 1414h | 1614h | Target Position | - |
| 22nd RPDO | 1415h | 1615h | Target Velocity | - |
| 23rd RPDO | 1416h | 1616h | Target Current | - |
| 24th RPDO | 1417h | 1617h | PVT Buffer | - |
| 1st TPDO | 1800h | 1A00h | StatusWord | - |
| 3rd TPDO | 1802h | 1A02h | StatusWord | Actual Position |
| 4th TPDO | 1803h | 1A03h | StatusWord | Actual Velocity |
| 5th TPDO | 1804h | 1A04h | StatusWord | Actual Current |
| 21st TPDO | 1814h | 1A14h | Actual Position | - |
| 22nd TPDO | 1815h | 1A15h | Actual Velocity | - |
| 23rd TPDO | 1816h | 1A16h | Actual Current | - |
| 24th TPDO | 1817h | 1A17h | PVT Buffer Position | - |
| 25th TPDO | 1818h | 1A18h | Prog. Digital Inputs | - |
| 26th TPDO | 1819h | 1A19h | Configurable. Contains 8 locations available for mapping objects. (See 1A19.01-1A19.08) | |

The relationship between a PDO Mapping parameter and Communication parameter is illustrated in Figure 15. The fact that PDO parameter objects are configured prior to any PDO messages being sent is what allows for all eight bytes of the PDO message to be used for data. The overall result is faster, more efficient data transfer and no additional bus usage for confirmation.

**Figure 15: PDO configuration parameters.**




### *4.9.4 The Communication Parameter Object*

The Communication Parameter object contains information regarding the COB-ID and transmission type of the PDO. The COB-ID and other settings are stored in sub-index 01h while the transmission type is stored in sub-index 02h. For example, the COB-ID of the 1st TPDO would be found at sub-index 1800.01h while the transmission type would be defined by sub-index 1800.02h. The details of choosing a COB-ID and setting the transmission type are explained below.

#### 4.9.4.1 SETTING COB-ID'S FOR EACH PDO

A unique COB-ID (unique with respect to the entire CANopen network, not just the node) must be assigned to each PDO which will be used over the CAN network. It is the system designer's responsibility to ensure that all PDOs have a unique COB-ID. It is best to assign the COB-IDs in a logical order, with the most important PDOs assigned to the lowest COB-IDs. The range of possible values is 181h-57Fh.

Sub-index 01h of each PDO's Communication Parameter object contains the COB-ID and is a 32-bit data field partitioned into five components as shown in Figure 16. Table 12 summarizes how these partitions are defined and Table 11 lists the object index for each PDO's Communication Parameter object.

**Figure 16: PDO COB-ID structure**

| Bit 31 | Bit 30 | Bit 29 | Bits 28 – 11 | Bits 10 – 0 |
|---|---|---|---|---|
| 0/1 | 0/1 | 0 | 000000000000000000 | COB-ID |

**Table 12: COB-ID bit definitions**

| Bit Number | Value | Description |
|---|---|---|
| 31(msb) | 0 | PDO message is enabled and will respond to the assigned trigger mechanism. |
| | 1 | PDO message is disabled and will not respond to the assigned trigger mechanism. This is the default state for all PDOs. |
| 30 | 0 | RTR allowed on this PDO. |
| | 1 | No RTR allowed on this PDO. |
| 29 | 0 | Use 0 for AMC drives (selects CAN 2.0A). |
| 28-11 | 0 | Use 0 for AMC drives (non-zero values reserved for CAN 2.0B). |
| 10-0 (lsb) | 11-bit Identifier | Holds the 11-bit identifier (COB-ID) of the PDO. Use the default value or set-up the priority for each PDO by setting this value closer to the value 181h, which has the highest PDO priority on a CAN network. |

#### 4.9.4.2 TRANSMISSION TYPE

Sub-index 02h of each PDO's Communication Parameter object is an 8-bit data field that defines the transmission type. Setting the value of this sub-index to an appropriate value, as given in Table 14, sets the transmission type. Note that there is a range of valid values for some transmission types. The "asynchronous" transmission type, for example, is set using a value of 254 or 255 (FEh or FFh).

**Table 13: PDO Transmission**

| TPDO Transmission Options | |
|---|---|
| Asynchronous | PDO transmission occurs based on specific event |
| Remotely Requested | Another device may request transmission of the PDO |
| Synchronous Transmission | The PDO may be transmitted (periodically or not) after expiration of a specified time period synchronized by reception of the SYNC message |
| **RPDO Execution Options** | |
| Asynchronous | Immediately upon receipt of the RPDO |
| Synchronous | Settings are applied after receipt of one or multiple SYNC messages. |

**Table 14: PDO Transmission Type selection table**

| PDO Transmission Description | | | |
|---|---|---|---|
| **Value** | **Transmission Type** | **TPDO** | **RPDO** |
| 01h – F0h | Synchronous Cyclic | PDO's are transmitted with relation to the Sync object. The number (01h-F0h) represents the number of Sync pulses between consecutive PDO transmissions. In addition, the PDO can be transmitted immediately following an RTR request. | The received data is held until the next Sync message. When the Sync message is received the data is applied |
| F1h - FBh | N/A | Reserved | Reserved |
| FCh | Synchronous RTR | PDO's are only transmitted following the first Sync message after a remote request. | Reserved |
| FDh | Asynchronous RTR | PDO's are only transmitted immediately following a remote request. | Reserved |
| FEh - FFh | Asynchronous | PDO's are transmitted immediately following an internal event or RTR request. | The received data is applied to its mapped objects immediately |
| | | | |

### *4.9.5 The Mapping Parameter Object*

The mapping parameter object contains information about each object mapped to a PDO. Each object that is mapped is represented by a sub-index in the Mapping Parameter object. So if, for example, a PDO has *n* number of mapped objects then the PDO's mapping parameter object will have sub-indices 1 through *n*. Each sub-index contains a 32-bit field partitioned into 3 components as shown in Figure 17.

**Figure 17: Mapping Parameter bit descriptions**

| Bits 31 – 16 | Bits 15 – 8 | Bits 7 – 0 |
|---|---|---|
| Index | Sub Index | Object Length |

The three components that represent a mapped object are described below:

- **Index:** The index of the object mapped to the PDO.
- **Sub-index:** The sub-index of the mapped object and the location of the data to be transmitted (zero if the object has no sub-indices).
- **Object Length:** The bit length (in hex) of the data to be transmitted. For example, 20h = 32 bits.

By placing information about an object in the Mapping Parameter, that object becomes mapped to the associated PDO. Mapping allows PDOs to know where they should read their data prior to transmission (in the case of TPDOs) or where they should write their data upon reception (in the case of RPDOs). Although DS301 allows up to 64 objects to be mapped to a single PDO, the number that can actually be mapped is ultimately determined by the total amount of the data mapped to the PDO. If, for example, a single object with an 8-byte (64-bit) data length is mapped to a PDO, then no other objects can be mapped to that same PDO since all 8-bytes of the data field will already be consumed. Mapped data is inserted into the data field of the PDO according to the order of mapping. That is, the data from the first mapped object consumes the first available byte (or bytes), and then data from the second mapped object consumes the next available byte (or bytes), and so on until all data bytes have been consumed or there is no more object data to map.

### 4.9.6 The RTR bit and TPDOs

Once a PDO has been configured and enabled, the host can use the RTR bit to request a TPDO from a node. This supplies the host with a fast and efficient on-demand method of retrieving information from a node. To request a TPDO, the host must send a message with the RTR bit set to 1 and a COB-ID that corresponds to the desired TPDO.

### 4.9.7 AMC PDO Assignment and Mapping

AMC CANopen drives support 8 RPDOs and 10 TPDOs, all of which can be assigned to a user-specified COB-ID. All 8 RPDOs are mapped to fixed, pre-defined objects and, as a result, only the Communication Parameter of an RPDO can be changed.

Similarly, all TPDOs, with the exception of TPDO 26, are mapped to fixed pre-defined objects and, again, only their Communication Parameters can be changed. The single exception, TPDO 26, is available for mapping up to 8 user specified application objects. All TPDOs can be assigned user-specified trigger mechanisms based on either timing or object data changes as explained in the following section. Some TPDOs, however, have fixed predefined trigger mechanisms. To know if a TPDO has a predefined trigger, check the description of that TPDO in the Object Dictionary.

### 4.9.8 AMC Asynchronous Transmission Events

AMC CANopen drives support 3 basic asynchronous event types:

- Time based: the drive transmits the selected TPDOs when a certain amount of time has elapsed. There are 2 internal timer objects available. Any of the TPDOs can be mapped to either or both timers.
- Value based: the drive monitors a certain object (presumably of a numerical type), and when the object has changed by a certain amount, the selected TPDOs will be transmitted. Two value counters exist, one watches for the mapped object to change by a specified amount, the other watches for the mapped object to reach a specific value. Any of the TPDOs can be mapped to either or both of the Value Counters.
- Bit based: the drive monitors a certain object (presumably of a bit-pattern type), and when a bit in that object changes (from 0 to 1 or 1 to 0), the selected TPDOs will be transmitted. Any of the TPDOs can be mapped to either or both of the Bit Watch processes.

The objects used to configure these asynchronous events, as well as some objects supplied for reading information about these events, are summarized in Table 15.

**Table 15 Asynchronous TPDO Transmission Events**

| Event Type | Event | Object Name | Object Index | Object Type |
|---|---|---|---|---|
| Time Based | Timer1 | TPDO Timer1 Cycle Time | 2120h | Configurable |
| | | TPDO Timer1 Assigned TPDOs | 2121h | Configurable |
| | | TPDO Timer1 Next Processing Time | 2122h | Informational |
| | Timer2 | TPDO Timer2 Cycle Time | 2123h | Configurable |
| | | TPDO Timer2 Assigned TPDOs | 2124h | Configurable |
| | | TPDO Timer2 Next Processing Time | 2125h | Informational |
| Value Based | Value-Changed | TPDO Value-Changed Object ID | 2130h | Configurable |
| | | TPDO Value-Changed Delta Value | 2131h | Configurable |
| | | TPDO Value-Changed Assigned TPDOs | 2132h | Configurable |
| | | TPDO Value-Changed Object Last Value | 2133h | Informational |
| | Value-Reached | TPDO Value-Reached Object ID | 2150h | Configurable |
| | | TPDO Value-Reached | 2151h | Configurable |
| | | TPDO Value-Reached Assigned TPDOs | 2152h | Configurable |
| | | TPDO Value-Reached Direction | 2153h | Configurable |
| Bit Based | Bits-Changed1 | TPDO Bits-Changed1 Object ID | 2140h | Configurable |
| | | TPDO Bits-Changed1 Object Bit Mask | 2141h | Configurable |
| | | TPDO Bits-Changed1 Assigned TPDOs | 2142h | Configurable |
| | | TPDO Bits-Changed1 Object Last Value | 2143h | Informational |
| | Bits-Changed2 | TPDO Bits-Changed1 Object ID | 2144h | Configurable |
| | | TPDO Bits-Changed1 Object Bit Mask | 2145h | Configurable |
| | | TPDO Bits-Changed1 Assigned TPDOs | 2146h | Configurable |
| | | TPDO Bits-Changed1 Object Last Value | 2147h | Informational |

Please refer to the Object Dictionary section for more details on these objects.

### *4.9.9 PDO Message Examples*

#### 4.9.9.1 PDO CONFIGURATION EXAMPLE

This example demonstrates using expedited SDO messages to configure two PDOs (there is no need to use segmented SDO's in this case because data is less than 4 bytes). Each PDO is enabled, assigned a COB-ID, and the trigger mechanisms set to an arbitrary mechanism.

| COB-ID | Number of Bytes | Message / Data | Description |
|---|---|---|---|
| 601 | 8 | 22 01 14 01 81 01 00 00 | Writing COB-ID 181 to 2nd RPDO (1401.01). Setting bit 32 here to 0 enables the PDO to be processed |
| 601 | 8 | 22 01 14 02 FE 00 00 00 | Setting trigger mechanism of 2nd RPDO (1401.02) to respond Immediately upon receipt of data. (See Table 14) |
| 601 | 8 | 22 14 18 01 85 01 00 00 | Writing COB-ID 185 to 21st TPDO (1814.01) Setting bit 32 here to 0 enables the PDO to be processed |
| 601 | 8 | 22 14 18 02 01 00 00 00 | Setting trigger mechanism of 21st TPDO (1814.01) to respond only upon receipt of a SYNC message. (See Table 14) |

| | | | |
|---|---|---|---|
| 000 | 2 | 01 01 | Sending NMT message to start node 1 communication state machine so that PDO messages may be processed. |
| 181 | 4 | 06 00 01 00 | Using 2nd RPDO to set the drive into Profile Position Mode and the Shutdown control state |
| 181 | 4 | 07 00 01 00 | Using 2nd RPDO to keep the drive in Profile Position Mode and set the Operation Disabled control state |
| 181 | 4 | 0F 00 01 00 | Using 2nd RPDO to keep the drive in Profile Position Mode and set the Operational Enabled control state |
| 80 | 1 | 00 | Start sending SYNC messages to cause the SYNC triggered TPDOs to send data to the host. |
| 185 | | FF FF FF FF | 21st TPDO response to SYNC message containing actual position = -1 counts |
| 80 | 1 | 00 | Next SYNC message from host |
| 185 | | 02 00 00 00 | 21st TPDO response to SYNC message containing actual position = 2 counts |
| 80 | 1 | 00 | Next SYNC message from host |
| 185 | | 05 00 00 00 | 21st TPDO response to SYNC message containing actual position = 5 counts |

#### 4.9.9.2 ASYNCHRONOUS TPDO TRANSMISSION EXAMPLE # 1

This example sets the timer1 event to 1000ms and assigns three TPDOs to transmit on every timer1 event. Prior to this example TPDOs have been assigned valid COB-IDs and are enabled.

| COB-ID | Number of Bytes | Message / Data | Description |
|---|---|---|---|
| 000 | 2 | 01 01 | Sending NMT message to start node 1 communication state machine so that PDO messages may be processed. |
| 601 | 8 | 22 20 21 00 E8 03 00 00 | Writing 1000 to object 2120.00. This sets the event timer to 1s intervals |
| 601 | 8 | 22 21 21 00 23 00 00 00 | Writing to bit-mask such that TPDOs 1, 3, and 22 are assigned to transmit according to the timer object |
| Wait 1000 ms | | | |
| 181 | 2 | 21 06 | 1st TPDO transmits after 1 second with it's data |
| 281 | 6 | 21 06 FE FF FF FF | 3rd TPDO transmits the same time as the 1st TPDO |
| 2C1 | 4 | 00 00 00 00 | 22nd TPDO transmits the same time as the 1st TPDO |
| 601 | 8 | 40 22 21 00 00 00 00 00 | Host sends SDO message to read 2122.00 for next timer1 event occurrence. |
| 581 | 8 | 42 22 21 00 B2 ED 97 02 | Node indicates next event occurs at 43511218 ms |
| Wait 1000 ms | | | |
| 181 | 2 | 21 06 | 1st TPDO transmits after 1 second with it's data |
| 281 | 6 | 21 06 FE FF FF FF | 3rd TPDO transmits the same time as the 1st TPDO |
| 2C1 | 4 | 00 00 00 00 | 22nd TPDO transmits the same time as the 1st TPDO |
| 601 | 8 | 40 22 21 00 00 00 00 00 | Host sends SDO message to read 2122.00 for next timer1 event occurrence. |
| 581 | 8 | 42 22 21 00 B2 ED 97 02 | Node indicates next event occurs at 43512218 ms |
| … | | | |
| 601 | 8 | 22 21 21 00 00 00 00 00 | Host writes to bit-mask such that no TPDOs are assigned to transmit. This stops the Timer1 event. |

#### 4.9.9.3 ASYNCHRONOUS TPDO TRANSMISSION EXAMPLE # 2

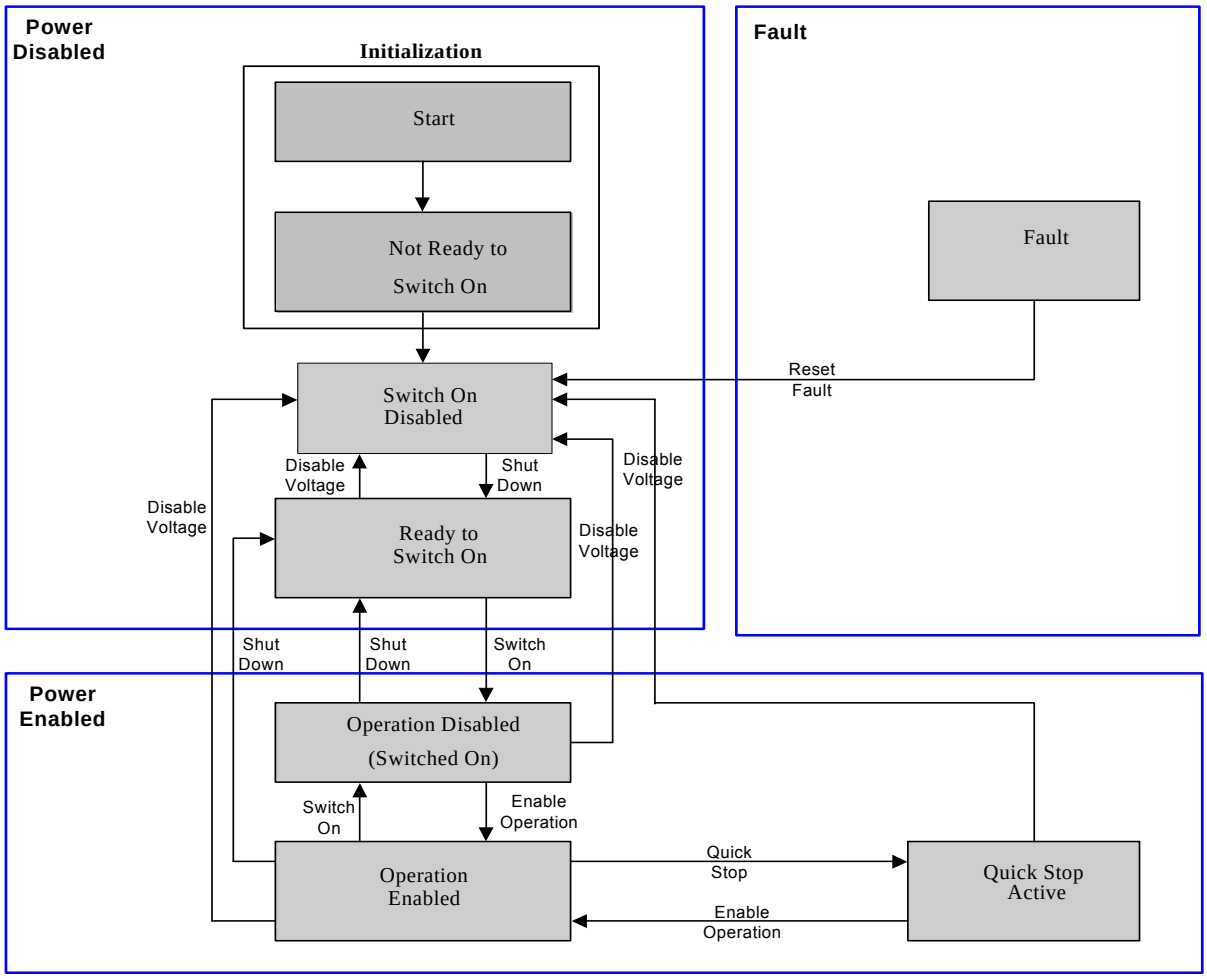
This example uses the bit based transmission events to monitor specific bits in the Actual Position object (6064h). Prior to this example TPDOs have been assigned valid COB-IDs and are enabled

| COB-ID | Number of Bytes | Message / Data | Description |
|---|---|---|---|
| 000 | 2 | 01 01 | Sending NMT message to start node 1 communication state machine so that PDO messages may be processed |
| 601 | 8 | 22 40 21 00 00 64 60 00 | Writing 60 64 00 to object 2140.00. This sets the Bit-Watch1 event to monitor object 6064h. Byte 8 is always 00 |
| 601 | 8 | 22 41 21 00 00 02 00 00 | Writing the exact bits to watch such that TPDOs will transmit when these/ this bit changes. This example watches bit 10 |
| 601 | 8 | 22 42 21 00 23 00 00 00 | Writing the Bit-mask to assign TPDOs 1, 3, and 22 to transmit on the bit change event |
| Wait until Bit 10 toggles | | | |
| 181 | 2 | 21 06 | 1st TPDO transmits after bit 10 toggle |
| 281 | 6 | 21 06 FE FF FF FF | 3rd TPDO transmits the same time as the 1st TPDO |
| 2C1 | 4 | 00 00 00 00 | 22nd TPDO transmits the same time as the 1st TPDO |
| 601 | 8 | 40 43 21 00 00 00 00 00 | Host sends SDO message to read 2143.00 for last value of monitored object. This is optional |
| 581 | 8 | 42 22 21 00 FE FF FF FF | Node indicates the last value contained -2 |
| Wait until Bit 10 toggles | | | |
| 181 | 2 | 21 02 | 1st TPDO transmits after bit 10 toggle |
| 281 | 6 | 21 02 00 00 00 00 | 3rd TPDO transmits the same time as the 1st TPDO |
| 2C1 | 4 | 4D 34 00 00 | 22nd TPDO transmits the same time as the 1st TPDO |
| 601 | 8 | 40 43 21 00 00 00 00 00 | Host sends SDO message to read 2143.00 for last value of monitored object. This is optional |
| 581 | 8 | 42 22 21 00 00 00 00 00 | Node indicates the last value contained 0 |
| … | | | |
| 601 | 8 | 22 42 21 00 00 00 00 00 | Host writes to bit-mask such that no TPDOs are assigned to transmit. This stops the Bit-Watch1 event |

## 5.1 State Machine Overview

CANopen drives operate based on a control state machine where each state has a defined behavior. The drive can be controlled to transition from one state to another in a particular order using the ControlWord object (6040h). This is a write only object used specifically to transition the drive's control state machine between states. Below is a graphical overview of the state machine. The grey boxes represent the states. The arrows represent the one-way path between states. The small text along the path of the arrow represents the command necessary to make each transition.

**Figure 18: ControlWord State machine block diagram.**




Upon power-up, the drive will automatically step through the 'Start' and 'Not Ready to Switch On' states, arriving at the 'Switch On Disabled' state. Further advancement to other states is accomplished by setting the ControlWord (Object index 6040) to the proper value. The commands that cause the state transitions in the state machine correspond to certain bit settings within the ControlWord. For example, to transfer from the 'Ready to Switch On' state to the 'Switched On State', one would use the Switch On command, by setting the ControlWord to the appropriate value (and hence bit pattern). The drive state may be queried by using SatusWord (Object index 6041). If the drive senses a fault, it will automatically move into the Fault state. The ControlWord can once again be used to move from the Fault state to the Not Ready to Switch On state.

## 5.2 Drive States

The following tables provide details on each of the CANopen states supported by AMC drives.

| Not Ready to Switch On | |
|---|---|
| Function | Part of drive initialization |
| Status | Logic Supply has been applied to the drive. The drive is being initialized. Drive functionality is disabled during this time. |
| Transitions | Transition to 'Switch On Disabled' is automatic when initialization complete. |

| Switch On Disabled | |
|---|---|
| Function | Drive initialization is complete. If a fatal error exists, the processor executes a Reset Fault command automatically. The drive is still disabled. |
| Status | Drive parameters have been set up. Only logic supply voltage is necessary at this time. Drive process monitoring may begin. |
| Transitions | Transition to the **Ready to Switch On** state is possible by a *Shut Down* command. |

| Ready to Switch On | |
|---|---|
| Function | Last state before Bridge enabled |
| Status | No energy is supplied to the motor. Control loops do not work. The drive function is still disabled. Bus power may be applied. |
| Transitions | Transition to **Operation Disabled (Switched ON)** state is possible via the *Switch On* command. Transition back to the **Switch On Disabled** state is possible via the *Disable Voltage* command, or by a *Quick Stop* command. |

| Operation Disabled (Switched On) | |
|---|---|
| Function | The bridge is turned on and a mode-dependent zero command is issued. |
| Status | The control loops are operational. Bus power is applied. The power section is switched on (if not already on). The target signal is not processed. The drive function is disabled. |
| Transitions | Transition to the **Operation Enabled** state is possible via the *Enable Operation* command. Transition back to the **Ready to Switch On** state is equally possible via the *Shut Down* command. Transition back to the **Switch On Disabled** state is possible via the *Disable Voltage* command or via a *Quick Stop* command. |

| Operation Enabled | |
|---|---|
| Function | This is the normal operation state of the drive. |
| Status | Power is supplied to the motor. Control loops are operational and target signals are processed. |
| Transitions | A *Quick Stop* command transfers the drive to the **Quick Stop Active** state. Transition back to the **Ready to Switch On** state is possible via the *Shut Down* command. Transition back to the **Switch On Disabled** state is possible via the *Disable Voltage* command or the *Drive Enable Input*. Transition back to the **Operation Disabled** state is possible via the *Switch On* command. |

| Quick Stop Active | |
|---|---|
| Function | The motor (shaft) is brought to a stop using the Quick Stop Ramp. |
| Status | Control loops are operational. Power is applied to the motor. The motor shaft is held in position in position mode or zero velocity in velocity mode. |
| Transitions | Transition back to the **Operation Enabled** state is possible via the *Enable Operation (7)* command. Transition back to the **Switch On Disabled** state is possible via *the Disable Voltage (4)* command, or via the *Drive Enable Input (2)* (both include the "Power Disable Delay" process). |

| Fault | |
|---|---|
| Function | A fault has occurred and has not yet been reset |
| Status | The power output stage is disabled; no energy is supplied to the motor. |
| Transitions | Transition to the Switch On Disabled state is possible via the *Reset Fault* command. |

## 5.3 ControlWord (6040h)

The following table shows the values used with object 6040h to cause transitions shown in figure 19 above. An example hexadecimal value is provided on the right.

**Table 16: ControlWord values**

| State Transition Command | Bit 7 | Bit 5 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Example Value |
|---|---|---|---|---|---|---|---|
| Reset Fault | 0→1 | X | X | X | X | X | 00 80 |
| Disable Voltage | 0 | X | X | X | 0 | X | XX 04 |
| Shutdown | 0 | X | X | 1 | 1 | 0 | XX 06 |
| Switch On<br>(Identical to Enable Operation) | 0 | X | 0 | 1 | 1 | 1 | XX 07 |
| Enable Operation | 0 | X | 1 | 1 | 1 | 1 | XX 0F |
| Quick Stop | 0 | X | X | 0 | 1 | X | XX 02 |
| Begin Homing<br>(Homing mode only) | 0 | 1 | 1 | 1 | 1 | 1 | XX 1F |
| End Homing<br>(Homing mode only) | 0 | 0 | 1 | 1 | 1 | 1 | XX 0F |
| 0 = OFF, 1 = ON, X = don't care | | | | | | | |

## 5.4 SatusWord (6041h)

The SatusWord reports exactly which state the drive is in. Table 17 defines each bit in the SatusWord and Table 18 shows how to interpret what state the drive is in via the combination of bits 1-3, 5 and 6. Each drive state is described in detail in Drive States.

**Table 17: SatusWord bit descriptions**

| Bits | Name | Descriptions |
|---|---|---|
| 0 | Ready to Switch On | See Table 18 to see how this bit relates to the control state machine. |
| 1 | Switched On | See Table 18 to see how this bit relates to the control state machine |
| 2 | Operation Enabled | See Table 18 to see how this bit relates to the control state machine |
| 3 | Fault | See Table 18 to see how this bit relates to the control state machine |
| 4 | Voltage Enabled | 1 when power is applied to the motor |
| 5 | Quick Stop | See Table 18 to see how this bit relates to the control state machine |
| 6 | Switch On disabled | See Table 18 to see how this bit relates to the control state machine |
| 7 | Warning | Object 205B can be used to configure which internal drive events will set this bit. |
| 8 | Manufacture specific | Object 205B can be used to configure which internal drive events will set this bit. |
| 9 | Remote | 1 when the drive is in remote mode. |
| 10 | Target Reached | 1 when the target is reached. (Position, Velocity, and Current modes) |
| 11 | Internal Limit Active | Object 205B can be used to configure which internal drive events will set this bit. |
| 12 | Homing complete | 1 when Homing completes, otherwise 0. |
| 13 | - | - |
| 14 | - | - |
| 15 | - | - |

**Table 18: SatusWord drive states**

| Drive State | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | SatusWord |
|---|---|---|---|---|---|---|---|---|
| Not Ready to Switch On | 0 | X | X | 0 | 0 | 0 | 0 | xxxx xxxx x0xx 0000 |
| Switch On Disabled | 1 | X | X | 0 | 0 | 0 | 0 | xxxx xxxx x1xx 0000 |
| Ready to Switch On | 0 | 1 | X | 0 | 0 | 0 | 1 | xxxx xxxx x01x 0001 |
| Switched On | 0 | 1 | X | 0 | 0 | 1 | 1 | xxxx xxxx x01x 0011 |
| Operation Enabled | 0 | 1 | X | 0 | 1 | 1 | 1 | xxxx xxxx x01x 0111 |
| Fault | 0 | X | X | 1 | 0 | 0 | 0 | xxxx xxxx x0xx 1000 |
| Quick Stop Active | 0 | 0 | X | 0 | 1 | 1 | 1 | xxxx xxxx x00x 0111 |
| 0 = OFF, 1 = ON, X = don't care | | | | | | | | |

# 6 HOMING

AMC CANopen drives support a wide variety of homing routines. These routines rely on signals such as limit switch, home switch, and encoder index signals to achieve precise starting positions. Four objects define the offset, speed, acceleration, and the particular homing method used. These objects are listed in the table below.

| Object Index | Description |
|---|---|
| 607Ch | Home Offset |
| 6099h | Homing Speeds |
| 609Ah | Homing Acceleration |
| 6098h | Homing Method |

## 6.1 Home Offset

The home offset specifies the difference between the home position and the zero position. The home position is the position of the motor when the home switch or encoder index is toggled during a homing routine. The zero position is the position defined to be zero as seen by the CAN master. If the home offset is set to zero, the home position will be equal to the zero position.

## 6.2 Homing Speeds

There are two homing speeds to take into consideration: the speed during the search for home switch, and the speed during the search for the index. Typically, the speed during the search for the home switch is set to be faster than the speed during the search for the index.

## 6.3 Homing Acceleration

A single value is used to define the acceleration and deceleration of all moves during the homing routine.

## 6.4 Homing Method

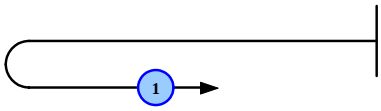
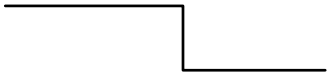
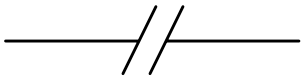
AMC CANopen homing methods depend on the presence of up to three different system components: an index pulse, a home switch, and a limit switch. The simplest homing methods require just one or none of these components, whereas the more complex methods require two or all of these components. All homing methods have been summarized in Table 19, along with their necessary components, and have been named according to [DSP402] which states that there are a total of 35 possible homing methods, some of which are reserved and not currently specified.

**Table 19**

| Homing Method | Index Pulse | Home Switch | Limit Switch |
|---|---|---|---|
| Methods 1 & 2 | ✓ | | ✓ |
| Methods 3 to 6 | ✓ | ✓ | |
| Methods 7 to 14 | ✓ | ✓ | ✓ |
| Methods 15 & 16 | Reserved | | |
| Methods 17 & 18 | | | ✓ |
| Methods 19 to 22 | | ✓ | |
| Methods 23 to 30 | | ✓ | ✓ |
| Methods 31 & 32 | Reserved | | |
| Methods 33 & 34 | ✓ | | |
| Method 35 | | | |

Because these homing methods can become fairly complex, they are best described visually. As a result, *homing diagrams* are utilized to illustrate the behavior of each method. Homing diagrams consist of multiple components each of which is described in Table 20.

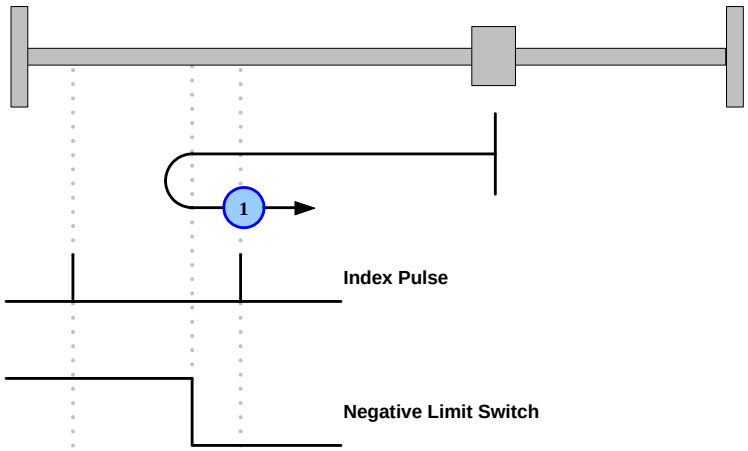
Table 20

| | |
|---|---|
| **Load and physical limits** |  |
| The square near the middle of the illustration shows the load object that is to be moved. The endpoints represent physical limitations or barriers, which the load cannot travel past. The left side is in the negative direction while the right side is in the positive direction. | |
| **Direction of travel** |  |
| The vertical line on the right side represents the starting position. The load travels in the direction of the arrow. In the illustration shown, the load begins traveling in the negative direction and then switches directions to move in the positive direction. The circle represents the home position at which point the (actual) measured position is reset to zero. The small section of arrow following the circle represents the distance traveled, past the home position, during deceleration of the load. Lastly, the number in the circle represents the number designated to that particular homing method. | |
| **Index Pulse** |  |
| Each vertical line represents one index pulse. | |
| **Limit/Home Switch** |  |
| A label in the actual homing diagram will be used to label a switch as either a limit/home switch. As shown, there are only two positions for a switch: high (active) or low (inactive). | |
| **Break** |  |
| Represents a break in the diagram. This is used for representing a length of distance too large to properly scale on the diagram. | |

### *6.4.1 Method 1: Homing on the Negative Limit Switch*

This method uses the negative limit switch and index to home the load. If the negative limit switch is off, the motor moves in the negative direction. Once the limit switch toggles, the motor changes direction and moves until the next encoder index. Homing is complete at this point. Figure 19 illustrates the homing diagram for this method.

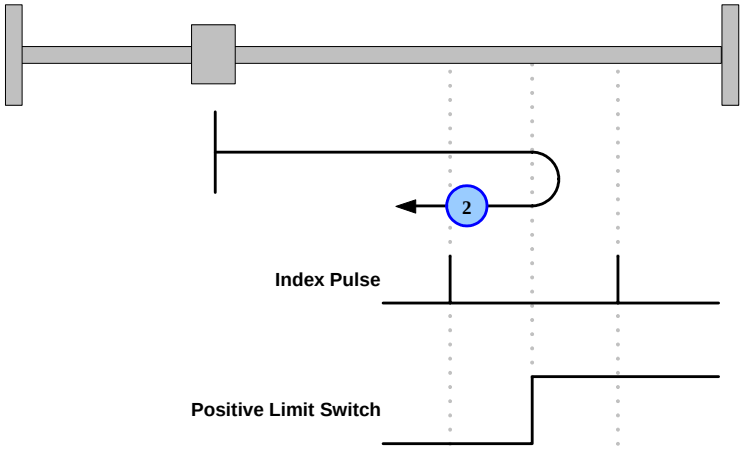
**Figure 19**




### *6.4.2 Method 2: Homing on the Positive Limit Switch*

This method uses the positive limit switch and index to home the load. If the positive limit switch is off, the motor moves in the positive direction. Once the limit switch toggles, the motor changes direction and moves until the next encoder index. Homing is complete at this point. Figure 20 illustrates the homing diagram for this method.

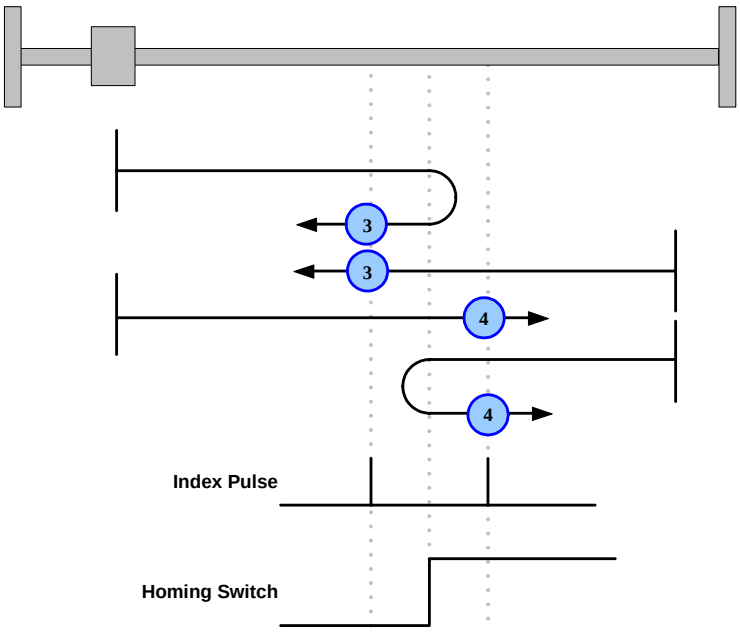
**Figure 20**




### *6.4.3 Methods 3 and 4: Homing on the Positive Home Switch*

These methods use the positive home switch and index to home the load. The initial direction of movement for a given routine method is dependent on the home switch position. However, the final position is always in the same direction. Homing methods 3 and four perform the same operations, but in opposite directions with opposite home switch polarity. Figure 21 illustrates the homing diagram for these methods.

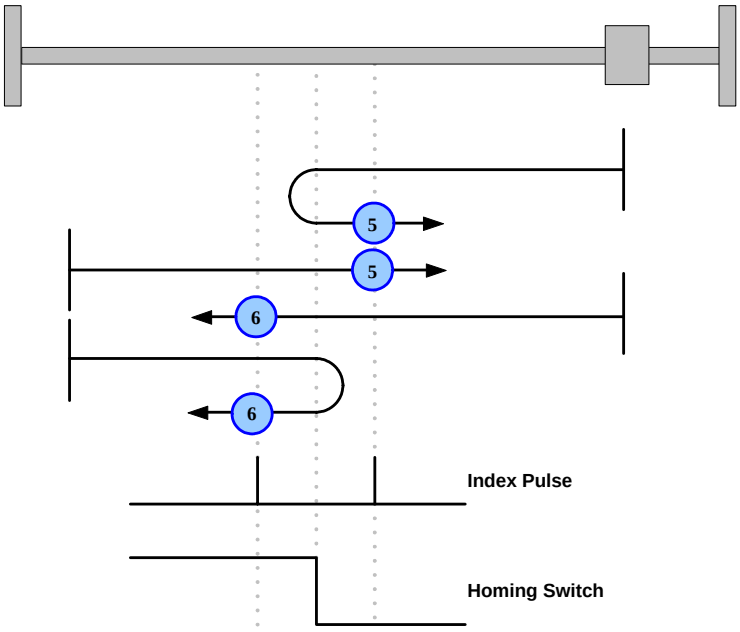
**Figure 21**




### 6.4.4 Methods 5 and 6: Homing on the Negative Home Switch

This is literally a mirror image of the homing routines used by methods 3 and 4. Figure 22 illustrates the homing diagram for these methods.

**Figure 22**

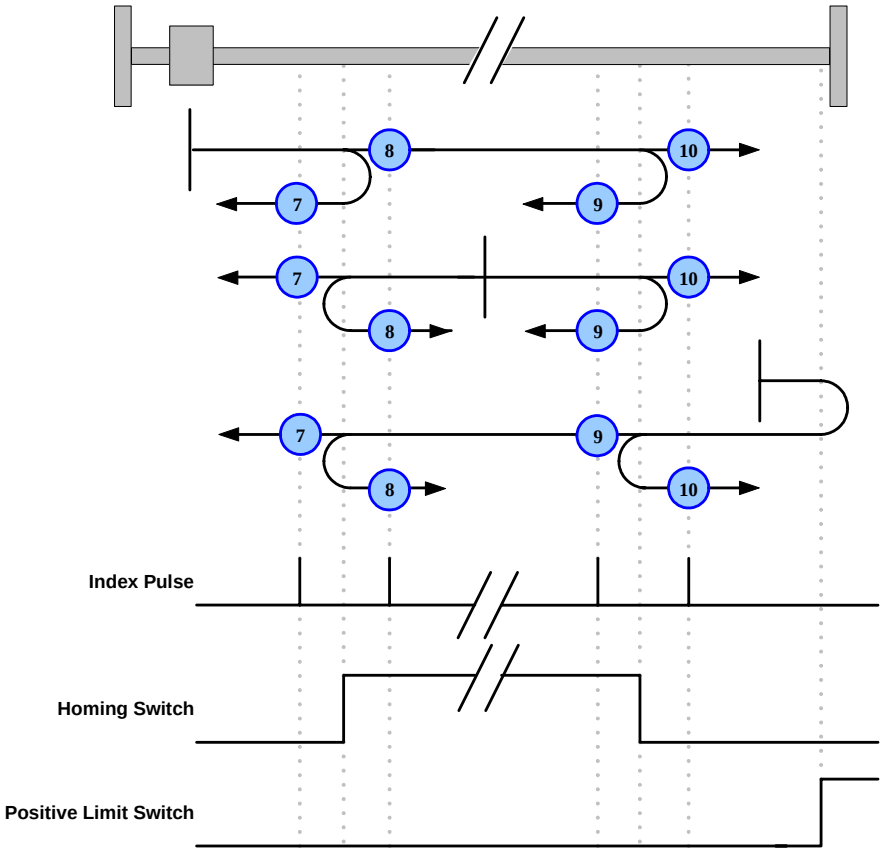



### 6.4.5 Methods 7-14: Homing on the Home Switch

These methods use all three possible homing components (index pulse, home switch, and limit switch) with the index pulse to the nearest right or left of the home switch always being the sought after home position. Methods 7 to 10 use a positive limit switch and if the starting position is outside the active home switch region the initial direction of travel is always positive. For cases where the starting position is inside the active home switch region the initial direction will depend upon the index pulse being sought after: methods 7 & 8 home

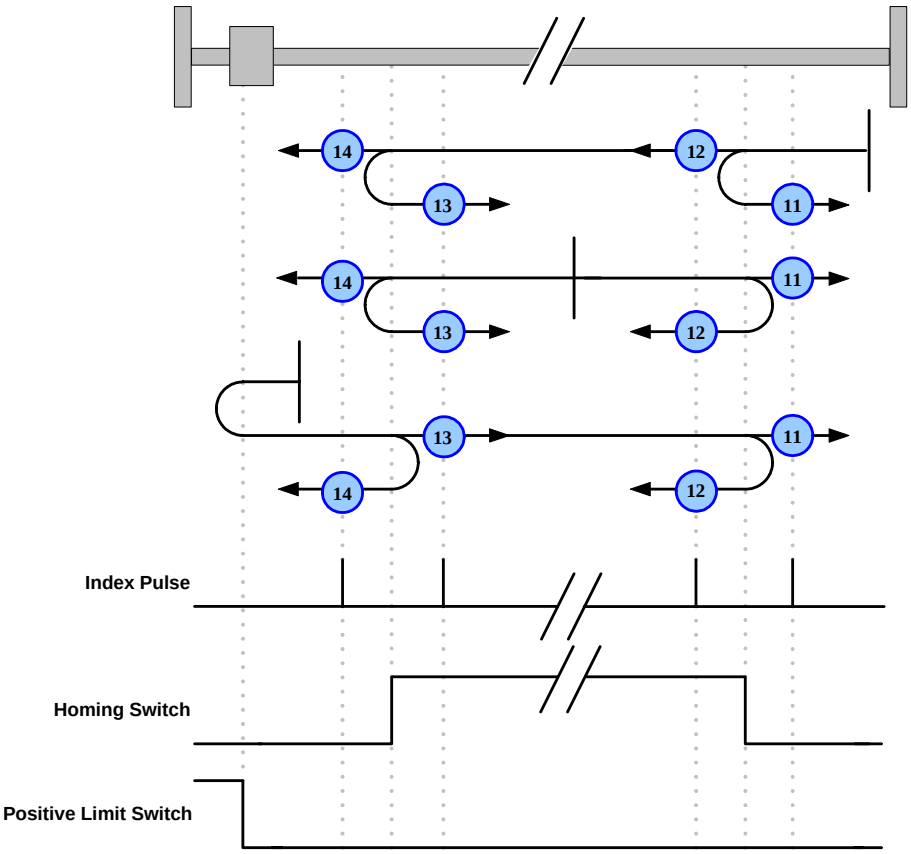
towards the left home switch edge so the initial direction will be left, whereas methods 9 & 10 home towards the right home switch edge so the initial direction will be right. Note that the only difference between methods 7 & 8 is that one homes to the index pulse left of the home switch edge whereas the other homes to the index pulse to the right; the same difference holds true for methods 9 & 10. Figure 23 illustrates the homing diagram for methods 7 to 10.

**Figure 23**




Methods 11 to 14 use a negative limit switch instead of a positive limit switch. As a result, the initial direction will be left, instead of right, whenever the starting point is outside of the active home switch region. Outside of this difference, methods 11 to 14 are identical to methods 7 to 10. Figure 24 illustrates the homing diagram for methods 11 to 14.

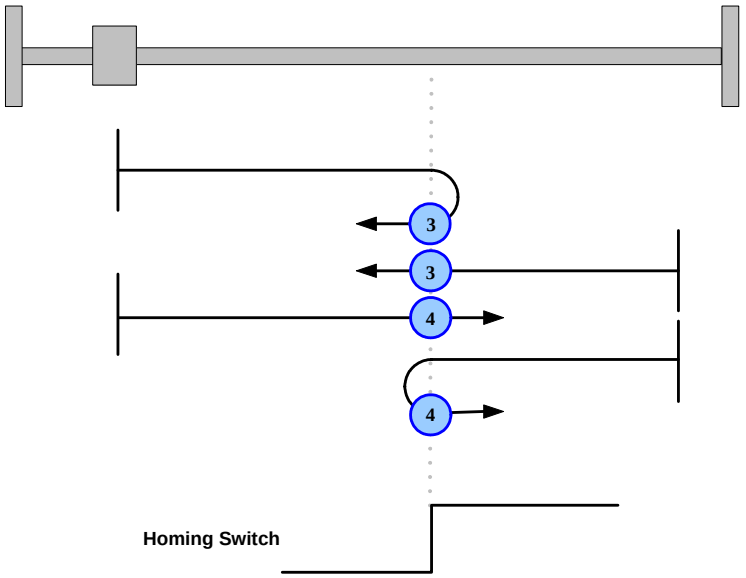
**Figure 24**




### *6.4.6 Methods 17-30: Homing without an index Pulse:*

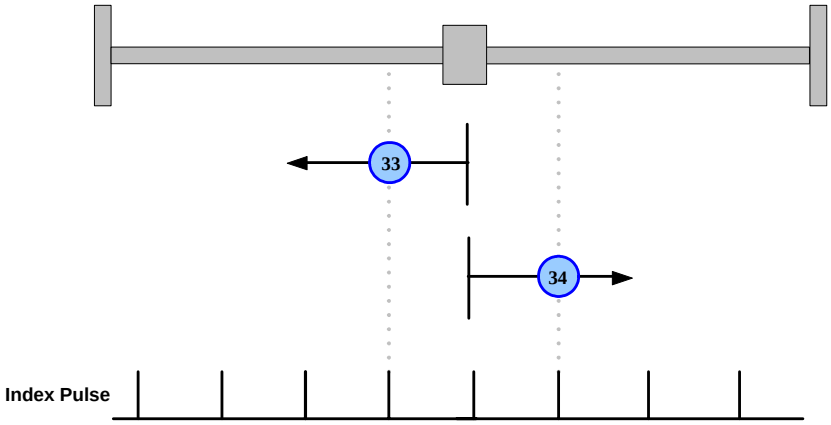
These homing routines use the same methods as 1 to 14, except the index pulse is not used. Instead, the home position is dependant on the edge of the relevant home or limit switch. To illustrate this difference, Figure 25 shows the homing diagram for methods 19 and 20, which are equivalent to methods 3 and 4 without the index pulse.

**Figure 25**



### 6.4.7 Methods 33 and 34: Homing on the Index pulse

These homing methods home to the nearest index pulse. Method 33 homes in the negative directions and method 34 homes in the positive direction.




### 6.4.8 Method 35

This homing method requires no index pulse or switches and involves nothing more than taking the current measured position to be home. This is accomplished by setting the actual measured position and target position to zero.

### 6.4.9 Homing Example

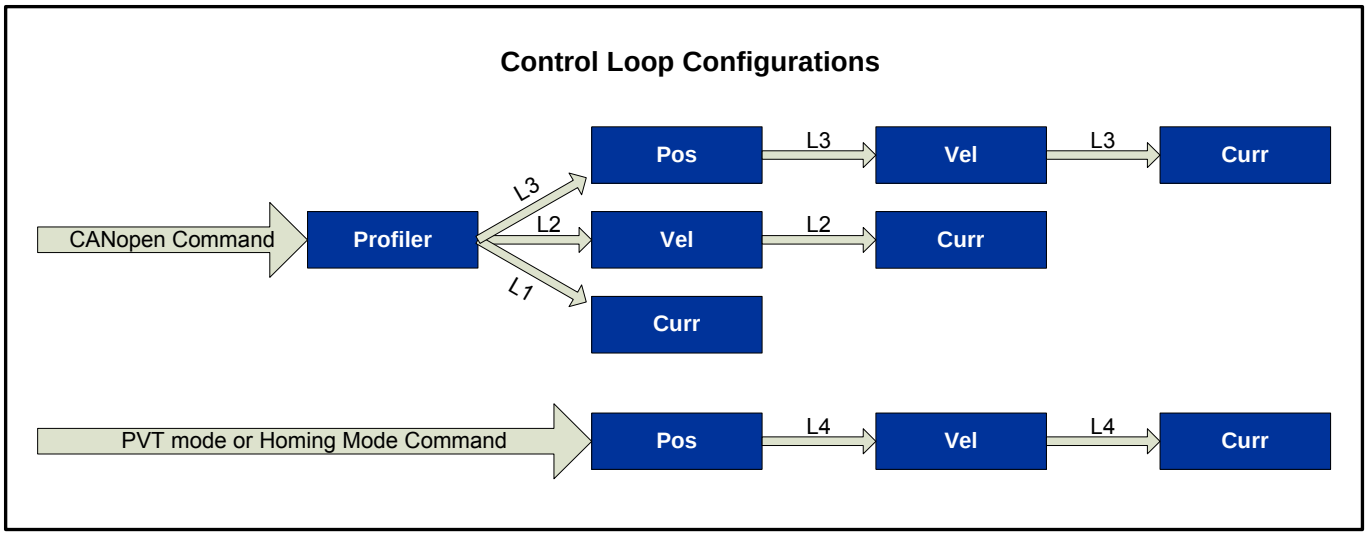
This example assumes the drive starts in Shutdown control state and Pre-Operational communication state. The 1st TPDO is setup to send upon any change in the StatusWord. The 13th bit of the StatusWord is the "Homing Complete" bit that will indicate when homing has completed and the drive mode may be changed.

| COB-ID | Number of Bytes | Message / Data | Description |
|---|---|---|---|
| 601 | 8 | 22 00 18 01 81 01 00 00 | Set 1st TPDO COB-ID to 181h |
| 601 | 8 | 22 00 18 02 FF 00 00 00 | Set 1st TPDO Trigger mechanism to "immediate" |
| 601 | 8 | 22 7C 60 00 00 00 00 00 | Write 0 to home offset object |
| 601 | 8 | 22 99 60 01 55 55 00 00 | Write 50 RPM to the Search For Home Switch speed |
| 601 | 8 | 22 99 60 02 55 55 00 00 | Write 50 RPM to the Search For Index Speed |
| 601 | 8 | 22 9A 60 00 37 89 41 00 | Write 10^5 Cnts/s^2 to Homing Acceleration |
| 601 | 8 | 22 98 60 00 22 00 00 00 | Set Homing to method 34, "home to index in positive direction" |
| 601 | 8 | 22 60 60 00 06 00 00 00 | Set the drive in Homing Mode |
| 000 | 2 | 01 01 | Start communication state machine so PDOs can be processed |
| 601 | 8 | 22 40 60 00 07 00 00 00 | Set node 1 to Operation Disabled |
| 601 | 8 | 22 40 60 00 0F 00 00 00 | Set node 1 to Operation Enabled |
| 601 | 8 | 22 40 60 00 1F 00 00 00 | Start Homing on node 1 |
| Wait for TPDO 1 to send a message containing 1 in the 13th bit. | | | |
| 601 | 8 | 22 40 60 00 0F 00 00 00 | Stop Homing on node 1 |
| 601 | 8 | 22 60 60 00 07 00 00 00 | Set node 1 in PVT mode |

AMC CANopen drives close position, velocity, and torque (current) loops that are configurable on the fly (via the CAN bus) during operation. There are 5 modes of operation available with object 6060h. Other modes of operation are achievable using the DriveWare300 software, however the drive cannot be commanded into these modes over the CAN bus. When changing loop configurations using object 6060h, velocity and position loop feedback sources are not touched. This means changing loop configurations assumes the feedback wiring and project parameters are configured properly for both the present loop and the one the drive is moving to.

Follow the formula for Expedited SDO messages in the "SDO" section of this manual when writing to object 6060h. More information on object 6060h is found in the Object Dictionary.

**Figure 26: Available loop configurations via CANopen messaging.**




## 7.1 Profile Current Mode: (L1 from Figure 26)

Presently AMC CANopen servo drives support Profile Current Mode, which is the basic building block of any CANopen servo system. The drives current loop consists of a PI loop. Because torque is merely a constant Kt multiplied by a magnitude of current, it is the programmer's responsibility to convert current values into torque values in the software environment.

The Command Profiler is enabled in this mode and sets limits on the rate of change of the current command, otherwise called Jerk. During a step acceleration command, Jerk is limited to the maximum value set in the profiler.

Tune this loop according to "current loop tuning" instructions in the appropriate DriveWare application help file.

The following objects are used to setup and operate the Current Mode:

| Object index | Name | Description |
|---|---|---|
| 6060h | Mode Of Operation | Sets the drive's mode of operation. |
| 203Ch | Command Profiler Parameters | Sets the values used by the command profiler to limit the target command. |
| 6086h | Motion Profile Type | Sets profiling to linear ramp. Currently this is fixed and read only. |
| 2010h | Current Values | Read instantaneous values such as Current Demand and Current Target. This object is read only. |
| 2034h | Current Loop and Commutation Values | Sets the tuning and commutation values associated with the current loop. |
| 6071h | Target Current | Sets the target current command in profile current mode. |

| | | |
|---|---|---|
| 6077h | Actual Current | Reads the actual motor current (in case of 3-phase motors, this is a composite, equivalent single phase current). |

## 7.2 Profile Velocity Mode: (L2 from Figure 26)

The AMC Velocity control loop is a fully de-coupled PID with an acceleration feedforward term, and a low speed estimator. In Profile Velocity Mode, the drive closes two control loops, velocity, and current. Velocity feedback may be derived from a motor mounted encoder or analog source with a 10V maximum. The low speed estimator is most useful when necessarily tight velocity loops can cause audible noise during low speed moves (less than 1 count per velocity update).

The Command Profiler is enabled in this mode. The Profiler sets limits on the rate of change of the velocity command, otherwise called acceleration. When commanding large velocity transients, the acceleration between points is limited to the maximum value set in the profiler.

When tuning the velocity loop, proportional gain is typically all that is needed. It is important, however, to start with a stable, yet responsive current loop. Feedforward gain can be added to improve tracking performance, if needed. More information on tuning is found in the DriveWare help files.

| Object index | Name | Description |
|---|---|---|
| 6060h | Mode Of Operation | Sets the drive's mode of operation. |
| 203Ch | Command Profiler Parameters | Sets the values used by the command profiler to limit the target command. |
| 6086h | Motion Profile Type | Sets profiling to linear ramp. Currently this is fixed and read only. |
| 2037h | Velocity Limits | Sets the trip points for various velocity events such as Over Speed. |
| 2036h | Velocity Loop Control Parameters | Sets the tuning values associated with the velocity loop |
| 2011h | Velocity Values | Read instantaneous values such as Velocity demand and Velocity Target. This object is read only. |
| 6069h | Velocity Sensor Actual Value | Same as 2011.01h, reads pre-filtered measured velocity value. |
| 606Bh | Velocity Demand | Same as 2011.04h, reads Velocity Demand value. |
| 606Ch | Actual Velocity | Same as 2011.02h, reads post-filtered measured velocity value. |
| 60FFh | Target Velocity | Sets the target velocity command in profile velocity mode. |

## 7.3 Profile Position Mode: (L3 from Figure 26)

The AMC Position control loop is a fully de-coupled PID with velocity and acceleration feedforward terms. In Profile Position Mode, the drive closes three control loops, position, velocity, and current. The velocity loop provides additional "stiffness," keeping the dynamic position errors minimal because the drive now reacts not only to position errors, but also to velocity errors (which can be interpreted as position error changes).

The Command Profiler is enabled in this mode. The Profiler sets limits on the rate of change of the position command, otherwise called velocity. When commanding point-to-point moves, the velocity between points is limited to the maximum value set in the profiler.

When tuning the position loop, proportional gain is typically all that is needed. It is important, however, to start with a stable, yet responsive velocity loop. Feedforward gain can be added to improve tracking performance, if needed. More information on tuning is found in the DriveWare application help files.

The following objects define how the drive will behave in Position mode.

| Object index | Name | Description |
|---|---|---|
| 6060h | Mode Of Operation | Sets the drive's mode of operation. |
| 203Ch | Command Profiler Parameters | Sets the values used by the command profiler to limit the target command. |

| | | |
|---|---|---|
| 6086h | Motion Profile Type | Sets profiling to linear ramp. Currently this is fixed and read only. |
| 2038h | Position Loop Control Parameters | Sets the tuning values associated with the position loop |
| 2039h | Position Limits | Sets the trip points for various position events such as Max Measured Position Limit. |
| 2012h | Position Values | Read instantaneous values such as Position demand and Position Target. This object is read only. |
| 6064h | Actual Position | Same as 2012.01h, reads measured position value. |
| 607Ah | Target Position | Sets the target position command in profile position mode. |

## 7.4 PVT (Interpolated Position Mode): (L4 from Figure 26)

PVT mode allows for synchronized multi axis move profiles using interpolated position and velocity. The three control loops, position, velocity, and current, are enabled while the profiler is disabled. The process for setting up and controlling motion using PVT Mode is explained in detail in PVT Mode.

## 7.5 Homing Mode: (L4 from Figure 26)

See Homing for detailed information about methods and hardware involved in homing.

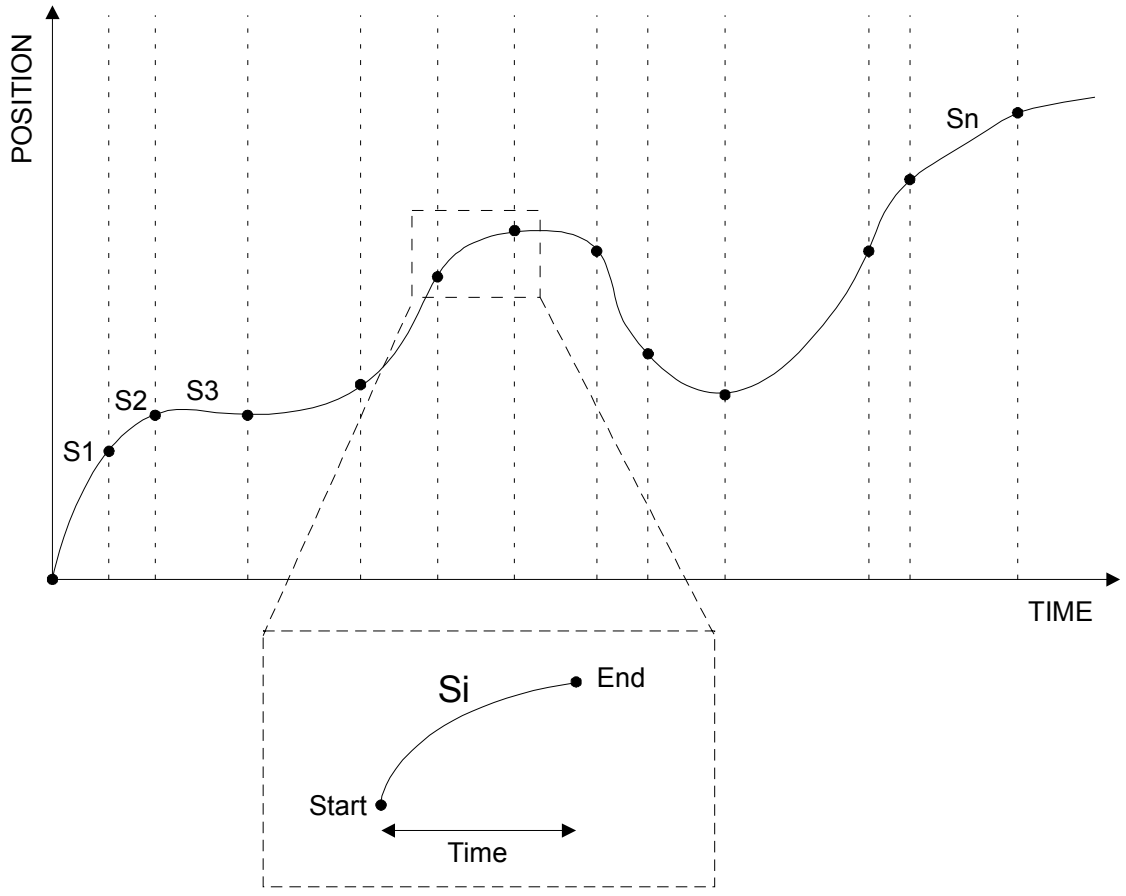
## 7.6 Non CANopen Defined Modes Of Operation

AMC digital servo drives provide flexibility beyond the CANopen defined standard modes of operation. For a case where a drive configuration is desired that is not available via object 6060h, contact AMC directly at http://www.a-m-c.com/download/form/form_productsupport.html for technical support.

## 8.1 PVT Overview

PVT mode is a position data-streaming mode that allows coordinated motion between multiple axes. Arbitrary position and velocity profiles can be executed on each axis. This is achieved via a so-called PVT command. A PVT command contains the position, velocity, and time information of profile segment end points. The servo drive performs a third order interpolation between segment end points. This results in a kind of partial trajectory generation where both host controller and servo drive generate a specific portion of the overall move profile trajectory. The host controller calculates position and velocity of intermittent points on the overall trajectory, while the servo drive interpolates between these intermittent points to ensure smooth motion. The actual position loop is closed within the drive. This reduces the amount of commands that need to be sent from host controller to drive, which is critical in distributed control systems. The number of segments and the time duration of each segment need to be selected based upon required accuracy and network bandwidth.

An arbitrary position profile can be split in multiple consecutive segments as follows:




Each segment has a start point and an end point. The end point of one segment is the start point of the next segment. Each segment end point (start or end) has a position and velocity value. The segment time can be variable depending on curvature (smaller time for rapidly changing positions).

PVT mode operates through PVT commands. A PVT command is an unconfirmed message (manufacturer specific RPDO 24). The PVT command contains segment end point position and velocity information, and segment time. A 15 level FIFO buffer alleviates host controller timing requirements. The buffer can be cleared and the buffer pointer can be re-positioned. The drive will also send the following PVT related error messages: buffer empty, buffer full, counter error, or message length error. The Time Stamp message can be used to maintain time synchronization of nodes involved in PVT motion.

## 8.2 PVT Messages

### *8.2.1 Enable PVT*

Since PVT commands are PDO messages, RPDO 24 must be enabled for PVT to work. To enable this PVT Buffer RPDO, configure its PDO Communication Parameter (1417.01h) to set bit 31 to 0 (enable PDO). In addition, the COB-ID for this PDO is selectable. Note that the following example assigns the COB-ID for this node to 531h.

| Arbitration Field | Data Field | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| COB-ID | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
| 600h + Node-ID | 22 | 17 | 14 | 01 | 31 | 05 | 00 | 00 |

### *8.2.2 Mode Selection*

To use PVT, the drive must be set for PVT Mode through Object 6060h (Modes of Operation). The message may look like this one where it is writing (without size indication) the value 07h for PVT mode into Object 6060h.

| Arbitration Field | Data Field | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| COB-ID | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
| 600h + Node-ID | 22 | 60 | 60 | 00 | 07 | 00 | 00 | 00 |

### *8.2.3 Configuration*

The following objects are useful for configuring the drive's behaviors in PVT mode. Set digital outputs to indicate PVT status or specify warning messages for minimum number of buffer points. When errors occur in PVT mode, select from multiple event actions to configure the drive to react appropriately.

| Object index | Sub-index Range | Name | Description |
|---|---|---|---|
| 2048h | 01h | PVT Parameters | Specifies the minimum number of buffered PVT end points before a warning message is sent |
| 205Ah | 31h – 35h | Digital Output Parameters | Assign digital outputs to indicate specific PVT status |
| 2064h | 1Ch – 20h | Fault Response Time Parameters | Sets the wait time before reacting to an occurrence of a PVT event |
| 2065h | 1Bh – 1Fh | Fault Event Action Parameters | Selects the event action when a PVT event occurs. Possible event actions include Disable Power Bridge, Dynamic Brake, and many others. |
| 2066h | 22h – 26h | Fault Recovery Time Parameters | Sets the amount of time after the cause of the PVT fault no longer exists before drive fault condition is cleared |
| 2067h | 1Fh – 23h | Fault Time-Out Window Parameters | Time after drive fault condition is cleared before a new occurrence is considered a new fault |
| 2068h | 27h – 2Bh | Fault Maximum Recoveries Parameters | Max number of faults before a permanent action is taken |

### *8.2.4 PVT Message Protocol*

Once the drive is configured, it is ready to receive PVT segment end points into its 15 level FIFO buffer. The construction of the PVT message is made up of the COB-ID and eight data bytes, which are made up of the segment end point position, velocity, segment time, and integrity counter. The COB-ID value is 530 + Node-ID. Note that both the Position and Velocity data bytes (three bytes each) are arranged in Little Endian format.

**Figure 27: PVT message construction**

| Arbitration Field | Data Field | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| COB-ID | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
| 530h + Node-ID | (LSB) Position Values (MSB) | | | (LSB) Velocity Values (MSB) | | | Time | Counter |

**Table 21: PVT message description**

| Data Bytes | Name | Description |
|---|---|---|
| Byte 1<br>Byte 2<br>Byte 3 | Position Segment End Point | The segment end point position is a 24-bit value in counts (absolute position). The data are entered as hexadecimal, where Byte 3 is the Most Significant Byte (MSB) and Byte 1 is the Least Significant Byte (LSB). |
| Byte 4<br>Byte 5<br>Byte 6 | Velocity Segment End Point | The segment end point velocity is a 24-bit value in counts per second multiplied by a scale factor of 0.0256. The data are entered as hexadecimal, where Byte 6 is the Most Significant Byte (MSB) and Byte 4 is the Least Significant Byte (LSB). |
| Byte 7 | Segment Time Duration | Time duration in milliseconds with a maximum of 255 (FFh) milliseconds |
| Byte 8 | Integrity Counter | The integrity counter is an incremental counter that starts at zero and wraps around after 255 (FFh). PVT commands with non-consecutive counter values will result in an error message. |

### *8.2.5 Clear Buffer*

If for any reason the PVT buffer should be cleared, write the value 01h to Object 60C4.06h will remove all the points previously loaded in the buffer. Byte 8, the counter, will need to start at 00 when loading the next buffer point.

| Arbitration Field | Data Field | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| COB-ID | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
| 600h + Node-ID | 22 | C4 | 60 | 06 | 01 | 00 | 00 | 00 |

### *8.2.6 End of Motion*

To end a PVT sequence, send a PVT command where the position bytes are the same as the previous position, the velocity bytes are zeros, and the time byte is also zero. The Integrity Counter, however, continues to increment. Here is an example of the last two PVT messages to end the sequence.

| Arbitration Field | Data Field | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| COB-ID | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
| 530h + Node-ID | P | P | P | V | V | V | T | C |

| Arbitration Field | Data Field | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| COB-ID | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
| 530h + Node-ID | P | P | P | 00 | 00 | 00 | 00 | C + 1 |

### *8.2.7 Start Motion*

Once there are enough PVT end points in the PVT buffer, motion may begin. With the drive in Operation Enabled state, sending a broadcast message with COB-ID 500h (no data bytes required) will start motion on all axes. Note that this command can be sent as soon as the nodes involved have received at least one PVT command. To ensure smooth motion, new PVT commands must be sent in a timely fashion.

| Arbitration Field | Data Field | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| COB-ID | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
| 500h | - | - | - | - | - | - | - | - |

### *8.2.8 Stop Motion*

When the drive executes the final PVT end sequence command, motion will stop. However as with any other modes, the ControlWord (Object 6040h) may stop the motion with a state change from the Operation Enabled state, to a disabled state such as Switch On Disabled.

| Arbitration Field | Data Field | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| COB-ID | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
| 600h + Node-ID | 22 | 40 | 60 | 00 | 04 | 00 | 00 | 00 |

## 8.3 PVT Status

The following objects display the PVT status of the drive.

| Object index | Sub-index range | Name | Description |
|---|---|---|---|
| 2002h | 06h | Drive Status | The bits in this sub-index provide status on the PVT buffer |
| 201Dh | 01h | PVT Status | Same as bits 0 – 5 of object 2002.06h |
| 201Dh | 02h | PVT Points Remaining | Remaining number of points in the buffer to be executed |
| 201Dh | 03h | PVT Sequence Number | The current PVT point in the buffer |

## 8.4 Buffer Characteristics

Object 60C4h is the Interpolation Data Configuration. It provides information regarding the PVT buffer and also allows modifications to the buffer, such as removing all the PVT end points already in the buffer.

| Object index | Sub-index range | Name | Description |
|---|---|---|---|
| 60C4h | 01h | Max Buffer Size | Maximum size of PVT buffer |
| 60C4h | 02h | Actual Buffer Size | Shows the actual size of the PVT buffer |
| 60C4h | 03H | Buffer Organization | Specifies that it is a FIFO buffer |
| 60C4h | 04H | Buffer Position | Indicates the position of the buffer |
| 60C4h | 05h | Size of Data Record | Indicates the length of a PVT point (8 bytes) |

| | | | |
|---|---|---|---|
| 60C4h | 06h | Buffer Clear | Clears all segment end points in the PVT buffer |

### *8.4.1 Error Messages*

The drive will generate error messages in PVT mode. The emergency message protocol (COB-ID 80h + Node-ID) is used to transmit the error message. Refer to EMERGENCY Messages for decoding emergency messages.

## 8.5 PVT Example

Assume size indication is used for SDO services. For a two-axis system with COB-ID for node 1 set to 531 and node 2 set to 532:

| COB-ID | Message / Data | Description |
|---|---|---|
| 601 | 2B 40 60 00 0E 00 00 00 | Shutdown command to node 1 |
| 602 | 2B 40 60 00 0E 00 00 00 | Shutdown command to node 2 |
| 601 | 2F 60 60 00 07 00 00 00 | Interpolated position mode (PVT), node 1 |
| 602 | 2F 60 60 00 07 00 00 00 | Interpolated position mode (PVT), node 2 |
| 601 | 2B 40 60 00 0F 00 00 00 | Enable operation command to node 1 |
| 602 | 2B 40 60 00 0F 00 00 00 | Enable operation command to node 2 |
| 531 | 01 02 03 00 00 00 80 00 | First segment end point position = 030201h<br>End point velocity = 0h,<br>Segment time is 128 ms, node 1 |
| 531 | 02 04 06 00 00 00 FF 01 | Second segment end point position = 060402h<br>End point velocity = 0h<br>Segment time is 255 ms, node 1 |
| 531 | 01 02 03 00 00 00 80 02 | Third segment end point position = 030201h<br>End point velocity = 0h<br>Segment time is 128 ms, node 1 |
| 531 | 01 02 03 00 00 00 00 03 | No more segments, node 1 |
| 532 | 01 02 03 00 00 00 80 00 | First segment end point position = 030201h<br>End point velocity = 0h<br>Segment time is 128 ms, node 2 |
| 532 | 02 04 06 00 00 00 FF 01 | Second segment end point position = 060402h<br>End point velocity = 0h<br>Segment time is 255 ms, node 2 |
| 532 | 01 02 03 00 00 00 80 02 | Third segment end point position = 30201h<br>End point velocity = 0h<br>Segment time is 128 ms, node 2 |
| 532 | 01 02 03 00 00 00 00 03 | No more segments, node 2 |
| 500 | - | Start motion on all nodes in PVT mode |

# 9 CONNECTING TO AN AMC CANOPEN DRIVE

Connecting to an AMC CANopen drive is possible via two communication interfaces on the drive. One interface is the CANopen communication interface, which is used after the drive is configured for proper operation. The other interface is an RS-232 serial communication interface. This is used when first configuring a drive project file according to the application needs and storing it to the drive's Non Volatile Memory.

## 9.1 RS-232 Interface Setup:

All that is needed is a standard serial cable (not a null-modem) connected from the drive RS-232 port to a computer. If the computer does not have a serial port on it, a converter such as USB to RS-232 may be used. Other converters exist and any of them, as long as they can operate between 9600 and 115200 baud will also work.

Higher baud rates will achieve better setup software performance when operating the oscilloscope an other various features. Refer to the drive manual and setup software help files for more information about connecting to the RS232 interface.

## 9.2 CAN Interface Setup:

Before communication can occur over a CANopen network, each node on the network must be configured for a specific node address, baud rate, and termination setting.

### *9.2.1 Node Addressing*

Each node in a CANopen network must have a unique Node-ID. Please refer to the hardware manual for information regarding address selection.

### *9.2.2 Baud Rate Selection*

Each node in a CANopen network (including the host) must operate at the same CAN bus bit rate. Please refer to the hardware manual for information regarding CAN bus baud rate selection.

### *9.2.3 Termination Setting*

The last node in a CANopen network must provide CAN bus termination. Please refer to the drive manual for information regarding termination options.

# 10 HARDWARE REQUIREMENTS

## 10.1 CAN Card

AMC CANopen drives communicate with any CAN compatible hardware. CAN hardware is readily available from a variety of vendors. PC based CAN controllers are found in several common forms such as parallel-to-CAN, USB-to-CAN, serial-to-CAN or PCI-to-CAN.

Regardless of manufacturer and type, the CAN controller must be installed along with its appropriate software.

## 10.2 API

Every CAN controller includes an API (application to programmer interface). This is a library of functions that allows a programmer to utilize the CAN card to communicate with nodes on a CANopen network. Documentation for the CAN card's API will be available from the manufacturer.

## 10.3 Mating Connector

AMC CANopen drives use a low-density, male, 9-pin D-SUB mating connector shown in the table below. All of the components can be obtained from AMP at www.amp.com, or by calling (800-522-6752).

| Parts Needed | Description | Part Number |
|---|---|---|
| D-SUB plug: | Main body, pins not inserted | 205204-4 |
| Shell Kit: | Outer shell, metal plated for shielding. Includes strain relief. | 748677-1 |
| Pins: | Insert pins for the Plug body. May be purchased loose or on a strip. | Loose: 5-66507-7<br>Strip: 3-66507-0 |

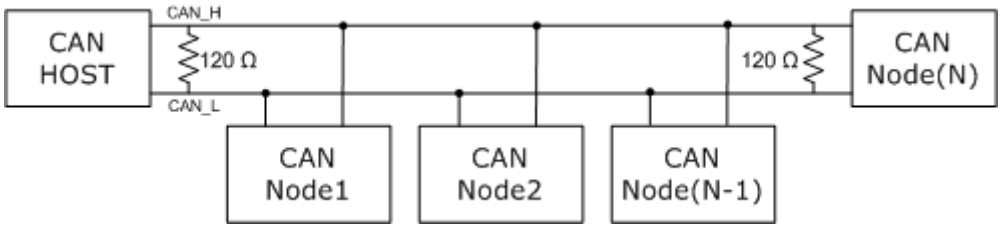
## 10.4 Wiring

Table 22 shows the standard AMC drive CANopen interface connector. Please note that the AMC ZDCR series drives have a different interface layout, refer to the drive's manual for a detailed description. Figure 28 shows an example of how the bus for an N node CANopen network should be wired.

**Table 22**

| PIN | NAME | Description | I/O |
|---|---|---|---|
| 1 | -- | Not Connected | NA |
| 2 | CAN_L | CAN_L bus line (dominant low) | Input |
| 3 | CAN_GND | CAN bus ground | GND |
| 4 | -- | Not Connected | NA |
| 5 | CAN_SHIELD | CAN shield | SHIELD |
| 6 | -- | Not Connected | NA |
| 7 | CAN_H | CAN_H bus line (dominant high) | Input |
| 8 | CAN_TERM | Termination. Connect to CAN_H for CAN bus termination via 120 Ohm resistor. | GND |
| 9 | CAN_V+ | Optional external supply (7.5 – 24 VDC) for communication | Input |

**Figure 28**




### *10.4.1 CAN_H, CAN_L, CAN_GND (Pins 7,2,3)*

These are a differential pair referenced to signal ground; they are considered the CAN bus.

### *10.4.2 CAN_V+ (Pin 9)*

Because the CAN interface can be completely isolated, external power may be required for the communication hardware in the drive. Please refer to the drive hardware manual for information regarding CAN interface isolation. The supply voltage common must connect to the CAN_GND, pin-3.

### *10.4.3 CAN SHIELD (Pin 5)*

AMC recommends using shielded cable with shielded twisted pairs. Each twisted pair should have one drain wire that must be terminated on one end only.

### *10.4.4 Proper Cable Shielding*

- Bring all twisted pair shields or drain wires to CAN_SHIELD, pin-5. Do not connect the shield to anything on the other end of the cable.
- Bring outer cable shield to the metal D-SUB connector shell that connects to the AMC drive. Do not connect the outer shield on the other end of the cable.

- DO NOT TERMINATE SHIELDS ON BOTH ENDS OF ANY CABLE; DOING SO WILL CREATE GOUND LOOPS AND POSSIBLY CREATE NOISE PROBLEMS!

### *10.4.5 CAN_TERM (Pin 8)*

The CAN network must be terminated by a 120 Ohm termination resistors on both ends. Generally the host controller will have the first 120-Ohm termination resistor in the network. The only other node to use a 120-Ohm termination resistor is the last node. Each node should branch from the main cable with the shortest possible stub length. This avoids reflections and transmission line effects in the communication line. If long branches are unavoidable, a termination resistor may be required.

# Object Dictionary

# 11 DICTIONARY TABLE FORMAT

The object dictionary provides one entry for each existing object. Since objects may or may not have sub-indices, the following convention is used for each entry:

**Figure 29: Object Table Example.**

| 2002.01h | Sub Index Name | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(15)}$ –1] | N/A | Read / Write* | No |
| **Description:**<br>Detailed description of how what this object does and how to use it.<br><br>* This indicates a note about conditions. | | | | |

In the example of Figure 29 the object index and sub-index is referenced via the dot (.). 2002h is the object index and .01h is the sub-index. Objects without sub-indices will be referenced without the dot (.).

Furthermore, each entry has the following attributes:

- Data Type: This field specifies the data type of the object. Data types can be 8-bit, 16-bit, 32-bit, or string.
- Range: This field specifies the usable range the values this object can contain.
- Units: This field specifies the units that apply to the value stored in this object. If the value contained in this object has no units, the field will contain “N/A.”
- Accessibility: This field specifies whether the object can be read or written to. If there is a * in this box, then the object may only be accessible in certain modes. See the Description box for more information about mode dependencies.
- Stored to NVM: This field specifies whether or not the object can be stored to Non Volatile Memory such that it is recalled on power up.
- Description: This field contains detailed information on the object and what it is used for.

## 12 CONFIGURATION OBJECTS

Although the following objects are used predominately during drive setup and initialization, they are not restricted to use only during setup. There are 3 sub-categories:

- Communication Objects: these objects determine the CANopen communication settings of the drive. They can only be set via the CANopen interface.
- Drive Objects: These objects are used to store to and restore from non-volatile memory, to read drive hardware and firmware information, and to set specific CANopen related options. They can only be used via the CANopen interface.
- Drive Configuration Objects: these objects determine the actual configuration of the drive, and their values are determined via the DriveWare300 setup software. The DriveWare300 application provides a convenient graphical environment that facilitates drive commissioning. **CAUTION: IT IS HIGHLY RECOMMENDED NOT TO MODIFY THESE OBJECTS AFTER THEY HAVE BEEN CONFIGURED WITH DRIVEWARE300!**

### 1000h: Device Type

| 1000h | Device Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | 0 – [$2^{(32)}$ –1] | N/A | Read Only | No |
| **Description:**<br>Contains information about the device type. This 32-bit object is split into two 16-bit fields. The first field describes the device profile and the second field supplies additional optional information about the device. AMC drives fit under device profile number 402 (Drives and Motion Control), which is represented by 0192h in the first 16-bit field. Servo drives are designated by setting the second bit of the second field (bit 17) to 1. | | | | |

### 100Bh: Stored Node-ID

| 100Bh | Stored Node-ID | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 1 – 127 | N/A | Read / Write | Yes |
| **Description:**<br>Stores the Node-ID assigned to the drive, when hardware settings are set for software addressing. | | | | |

### 100Ch: Guard Time

| 100Ch | Guard Time | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |
| **Description:**<br>Used with object 100Dh (Life Time Factor) to store the guard time in ms and the Life Time Factor. The Life Time Factor multiplied with the guard time gives the lifetime for the Life Guarding Protocol. | | | | |

### 100Dh: Life Time Factor

| 100Dh | Life Time Factor | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 – [$2^{(8)}$ –1] | N/A | Read / Write | Yes |

| Description: |
|---|
| Used with object 100Ch (Guard Time) to store the guard time in ms and the Life Time Factor. The Life Time Factor multiplied with the guard time gives the lifetime for the Life Guarding Protocol. |

### 1018h: Identity Object

| 1018.01h | Vendor ID | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>A unique vendor identifier. Always BDh for AMC drives. | | | | |

### 1400h: 1st Receive PDO Communication Parameter

This PDO is valid in all operating modes. The user can set the COB-ID of this PDO to any value. See object 1600h for details about the data transmitted by this PDO.

| 1400.01h | COB-ID Used By PDO | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | Yes |
| **Description:**<br>Holds the COB-ID of the PDO as well as other parameters. For details see Section 4.9.4.1. | | | | |

| 1400.02h | Transmission Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 – 255 | N/A | Read / Write | Yes |
| **Description:**<br>Defines the way in which the PDO will be transmitted, namely synchronous or asynchronous. For details see Section 4.9.4.2. | | | | |

### 1600h: 1st Receive PDO Mapping Parameter

This PDO is used to set the state of the drive (ex: ready, not ready, enabled, disabled, etc.). The object mapped to this PDO is fixed and not user selectable. See object 1400h for details on the transmission method.

| 1600.01h | PDO Mapping for the 1st Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the ControlWord object (6040h). For details about the format of this sub-index see Section 4.9.5. | | | | |

### 1401h: 2nd Receive PDO Communication Parameter

This PDO is valid in all operating modes. The user can set the COB-ID of this PDO to any value. See object 1601h for details about the data transmitted by this PDO.

| 1401.01h | COB-ID Used By PDO | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | Yes |
| **Description:**<br>Holds the COB-ID of the PDO as well as other parameters. For details see Section 4.9.4.1. | | | | |

| 1401.02h | Transmission Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 – 255 | N/A | Read / Write | Yes |
| **Description:**<br>Defines the way in which the PDO will be transmitted, namely synchronous or asynchronous. For details see Section 4.9.4.2. | | | | |

### 1601h: 2nd Receive PDO Mapping Parameter

This PDO is used to set both the state of the drive (ex: enabled, disabled, faulted, etc.) and the mode of operation of the drive (ex: torque, velocity, or position modes). The objects mapped to this PDO are fixed and not user selectable. See object 1401h for details on the transmission method.

| 1601.01h | PDO Mapping for the 1st Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the ControlWord object (6040h). For details about the format of this sub-index see Section 4.9.5. | | | | |

| 1601.02h | PDO Mapping for the 2nd Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the Modes of Operation object (6060h). For details about the format of this sub-index see Section 4.9.5. | | | | |

### 1402h: 3rd Receive PDO Communication Parameter

This PDO is valid in profile position mode only and does not exist in other modes. The user can set the COB-ID of this PDO to any value. See object 1602h for details about the data transmitted by this PDO.

| 1402.01h | COB-ID Used By PDO | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | Yes |
| **Description:**<br>Holds the COB-ID of the PDO as well as other parameters. For details see Section 4.9.4.1. | | | | |

| 1402.02h | Transmission Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 – 255 | N/A | Read / Write | Yes |
| **Description:**<br>Defines the way in which the PDO will be transmitted, namely synchronous or asynchronous. For details see Section 4.9.4.2. | | | | |

### 1602h: 3rd Receive PDO Mapping Parameter

This PDO is used to set both the state of the drive (ex: enabled, disabled, faulted, etc.) and the target position of the drive. The PDO is only available in profile position mode (see object 6060h for operating modes). The objects mapped to this PDO are fixed and not user selectable. See object 1402h for details on the transmission method.

| 1602.01h | PDO Mapping for the 1st Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the ControlWord object (6040h). For details about the format of this sub-index see Section 4.9.5. | | | | |

| 1602.02h | PDO Mapping for the 2nd Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the Target Position object (607Ah). For details about the format of this sub-index see Section 4.9.5. | | | | |

### 1403h: 4th Receive PDO Communication Parameter

This PDO is valid in profile velocity mode only and does not exist in other modes. The user can set the COB-ID of this PDO to any value. See object 1603h for details about the data transmitted by this PDO.

| 1403.01h | COB-ID Used By PDO | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | Yes |
| **Description:**<br>Holds the COB-ID of the PDO as well as other parameters. For details see Section 4.9.4.1. | | | | |

| 1403.02h | Transmission Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 – 255 | N/A | Read / Write | Yes |
| **Description:**<br>Defines the way in which the PDO will be transmitted, namely synchronous or asynchronous. For details see Section 4.9.4.2. | | | | |

### 1603h: 4th Receive PDO Mapping Parameter

This PDO is used to set both the state of the drive (ex: enabled, disabled, faulted, etc.) and the target velocity of the drive. The PDO is only available in profile velocity mode (see object 6060h for operating modes). The objects mapped to this PDO are fixed and not user selectable. See object 1403h for details on the transmission method.

| 1603.01h | PDO Mapping for the 1st Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the ControlWord object (6040h). For details about the format of this sub-index see Section 4.9.5. | | | | |

| 1603.02h | PDO Mapping for the 2nd Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the Target Velocity object (60FFh). For details about the format of this sub-index see Section 4.9.5. | | | | |

### 1404h: 5th Receive PDO Communication Parameter

This PDO is valid in profile torque mode only and does not exist in other modes. The user can set the COB-ID of this PDO to any value. See object 1604h for details about the data transmitted by this PDO.

| 1404.01h | COB-ID Used By PDO | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | Yes |
| **Description:**<br>Holds the COB-ID of the PDO as well as other parameters. For details see Section 4.9.4.1. | | | | |

| 1404.02h | Transmission Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 – 255 | N/A | Read / Write | Yes |
| **Description:**<br>Defines the way in which the PDO will be transmitted, namely synchronous or asynchronous. For details see Section 4.9.4.2. | | | | |

### 1604h: 5th Receive PDO Mapping Parameter

This PDO is used to set both the state of the drive (ex: enabled, disabled, faulted, etc.) and the target torque of the drive. The PDO is only available in profile torque mode (see object 6060h for operating modes). The objects mapped to this PDO are fixed and not user selectable. See object 1404h for details on the transmission method.

| 1604.01h | PDO Mapping for the 1st Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the ControlWord object (6040h). For details about the format of this sub-index see Section 4.9.5. | | | | |

| 1604.02h | PDO Mapping for the 2nd Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the Target Current object (6071h). For details about the format of this sub-index see Section 4.9.5. | | | | |

### 1414h: 21st Receive PDO Communication Parameter

This PDO is valid in profile position mode only and does not exist in other modes. The user can set the COB-ID of this PDO to any value. See object 1614h for details about the data transmitted by this PDO.

| 1414.01h | COB-ID Used By PDO | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | Yes |
| **Description:**<br>Holds the COB-ID of the PDO as well as other parameters. For details see Section 4.9.4.1. | | | | |

| 1414.02h | Transmission Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 – 255 | N/A | Read / Write | Yes |

| |
|---|
| **Description:**<br>Defines the way in which the PDO will be transmitted, namely synchronous or asynchronous. For details see Section 4.9.4.2. |

### 1614h: 21st Receive PDO Mapping Parameter

This PDO is used to set the target position of the drive. The PDO is only available in profile position mode (see object 6060h for operating modes). The object mapped to this PDO is fixed and not user selectable. See object 1414h for details on the transmission method.

| 1614.01h | PDO Mapping for the 1st Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the Target Position object (607Ah). For details about the format of this sub-index see Section 4.9.5. | | | | |

### 1415h: 22nd Receive PDO Communication Parameter

This PDO is valid in profile velocity mode only and does not exist in other modes. The user can set the COB-ID of this PDO to any value. See object 1615h for details about the data transmitted by this PDO.

| 1415.01h | COB-ID Used By PDO | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | Yes |
| **Description:**<br>Holds the COB-ID of the PDO as well as other parameters. For details see Section 4.9.4.1. | | | | |

| 1415.02h | Transmission Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 – 255 | N/A | Read / Write | Yes |
| **Description:**<br>Defines the way in which the PDO will be transmitted, namely synchronous or asynchronous. For details see Section 4.9.4.2. | | | | |

### 1615h: 22nd Receive PDO Mapping Parameter

This PDO is used to set the target velocity of the drive. The PDO is only available in profile velocity mode (see object 6060h for operating modes). The object mapped to this PDO is fixed and not user selectable. See object 1415h for details on the transmission method.

| 1615.01h | PDO Mapping for the 1st Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the Target Velocity object (60FFh). For details about the format of this sub-index see Section 4.9.5. | | | | |

### 1416h: 23rd Receive PDO Communication Parameter

This PDO is valid in profile torque mode only and does not exist in other modes. The user can set the COB-ID of this PDO to any value. See object 1616h for details about the data transmitted by this PDO.

| 1416.01h | COB-ID Used By PDO | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | Yes |
| **Description:**<br>Holds the COB-ID of the PDO as well as other parameters. For details see Section 4.9.4.1. | | | | |

| 1416.02h | Transmission Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 – 255 | N/A | Read / Write | Yes |
| **Description:**<br>Defines the way in which the PDO will be transmitted, namely synchronous or asynchronous. For details see Section 4.9.4.2. | | | | |

### 1616h: 23rd Receive PDO Mapping Parameter

This PDO is used to set the target current of the drive. The PDO is only available in profile torque mode (see object 6060h for operating modes). The object mapped to this PDO is fixed and not user selectable. See object 1416h for details on the transmission method.

| 1616.01h | PDO Mapping for the 1st Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the Target Current object (6071h). For details about the format of this sub-index see Section 4.9.5. | | | | |

### 1417h: 24th Receive PDO Communication Parameter

This PDO is valid in interpolated position mode (PVT mode) only and does not exist in other modes. The user can set the COB-ID of this PDO to any value. See object 1617h for details about the data transmitted by this PDO.

| 1417.01h | COB-ID Used By PDO | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | Yes |
| **Description:**<br>Holds the COB-ID of the PDO as well as other parameters. For details see Section 4.9.4.1. | | | | |

| 1417.02h | Transmission Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 – 255 | N/A | Read / Write | Yes |
| **Description:**<br>Defines the way in which the PDO will be transmitted, namely synchronous or asynchronous. For details see Section 4.9.4.2. | | | | |

### 1617h: 24th Receive PDO Mapping Parameter

This PDO is used to send PVT commands (set-points) to the drive. The PDO is only available in interpolated position mode (see object 6060h for operating modes). The object mapped to this PDO is fixed and not user selectable. See object 1417h for details on the transmission method.

| 1617.01h | PDO Mapping for the 1st Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the Interpolation Data Record object (60C1h). For details about the format of this sub-index see Section 4.9.5. | | | | |

### 1800h: 1st Transmit PDO Communication Parameter

This PDO is transmitted upon a user configurable event (see objects 2120h – 2125h, 2130h – 2133h, 2140h – 2147h and 2150h – 2153h), can be transmitted upon a SYNC message or when an RTR is received if the sub-indices of this object are configured appropriately. See object 1A00h for details about the data transmitted by this PDO.

| 1800.01h | COB-ID Used By PDO | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | No |
| **Description:**<br>Holds the COB-ID of the PDO as well as other parameters. For details see Section 4.9.4.1. | | | | |

| 1800.02h | Transmission Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 – 255 | N/A | Read / Write | No |
| **Description:**<br>Defines the way in which the PDO will be transmitted, namely synchronous or asynchronous. For details see Section 4.9.4.2. | | | | |

### 1A00h: 1st Transmit PDO Mapping Parameter

This PDO transmits drive status information. The object mapped to this PDO is fixed and not user selectable. See object 1800h for details on the transmission method.

| 1A00.01h | PDO Mapping for the 1st Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the StatusWord object (6041h). For details about the format of this sub-index see Section 4.9.5. | | | | |

### 1802h: 3rd Transmit PDO Communication Parameter

This PDO is transmitted upon a user configurable event (see objects 2120h – 2125h, 2130h – 2133h, 2140h – 2147h and 2150h – 2153h), can be transmitted upon a SYNC message or when an RTR is received if the sub-indices of this object are configured appropriately. See object 1A02h for details about the data transmitted by this PDO.

| 1802.01h | COB-ID Used By PDO | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | No |
| **Description:**<br>Holds the COB-ID of the PDO as well as other parameters. For details see Section 4.9.4.1. | | | | |

| 1802.02h | Transmission Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 – 255 | N/A | Read / Write | No |
| **Description:**<br>Defines the way in which the PDO will be transmitted, namely synchronous or asynchronous. For details see Section 4.9.4.2. | | | | |

### 1A02h: 3rd Transmit PDO Mapping Parameter

This PDO transmits drive status information and the actual position value stored in the drive. The objects mapped to this PDO are fixed and not user selectable. See object 1802h for details on the transmission method.

| 1A02.01h | PDO Mapping for the 1st Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the StatusWord object (6041h). For details about the format of this sub-index see Section 4.9.5. | | | | |

| 1A02.02h | PDO Mapping for the 2nd Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | 0 – 255 | N/A | Read Only | No |
| **Description:**<br>Maps the Actual Position Value object (6064h). For details about the format of this sub-index see Section 4.9.5. | | | | |

### 1803h: 4th Transmit PDO Communication Parameter

This PDO is transmitted upon a user configurable event (see objects 2120h – 2125h, 2130h – 2133h, 2140h – 2147h and 2150h – 2153h), can be transmitted upon a SYNC message or when an RTR is received if the sub-indices of this object are configured appropriately. See object 1A03h for details about the data transmitted by this PDO.

| 1803.01h | COB-ID Used By PDO | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | No |
| **Description:**<br>Holds the COB-ID of the PDO as well as other parameters. For details see Section 4.9.4.1. | | | | |

| 1803.02h | Transmission Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 – 255 | N/A | Read / Write | No |
| **Description:**<br>Defines the way in which the PDO will be transmitted, namely synchronous or asynchronous. For details see Section 4.9.4.2. | | | | |

### 1A03h: 4th Transmit PDO Mapping Parameter

This PDO transmits drive status information and the actual velocity value stored in the drive. The objects mapped to this PDO are fixed and not user selectable. See object 1803h for details on the transmission method.

| 1A03.01h | PDO Mapping for the 1st Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |

| | | | | |
|---|---|---|---|---|
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the StatusWord object (6041h). For details about the format of this sub-index see Section 4.9.5. | | | | |

| 1A03.02h | PDO Mapping for the 2nd Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | 0 – 255 | N/A | Read Only | No |
| **Description:**<br>Maps the Actual Velocity Value object (606Ch). For details about the format of this sub-index see Section 4.9.5. | | | | |

### 1804h: 5th Transmit PDO Communication Parameter

This PDO is transmitted upon a user configurable event (see objects 2120h – 2125h, 2130h – 2133h, 2140h – 2147h and 2150h – 2153h), can be transmitted upon a SYNC message or when an RTR is received if the sub-indices of this object are configured appropriately. See object 1A04h for details about the data transmitted by this PDO.

| 1804.01h | COB-ID Used By PDO | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | No |
| **Description:**<br>Holds the COB-ID of the PDO as well as other parameters. For details see Section 4.9.4.1. | | | | |

| 1804.02h | Transmission Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 – 255 | N/A | Read / Write | No |
| **Description:**<br>Defines the way in which the PDO will be transmitted, namely synchronous or asynchronous. For details see Section 4.9.4.2. | | | | |

### 1A04h: 5th Transmit PDO Mapping Parameter

This PDO transmits drive status information and the actual torque value stored in the drive. The objects mapped to this PDO are fixed and not user selectable. See object 1804h for details on the transmission method.

| 1A04.01h | PDO Mapping for the 1st Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the StatusWord object (6041h). For details about the format of this sub-index see Section 4.9.5. | | | | |

| 1A04.02h | PDO Mapping for the 2nd Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the Actual Current Value object (6077h). For details about the format of this sub-index see Section 4.9.5. | | | | |

### 1814h: 21st Transmit PDO Communication Parameter

This PDO is transmitted upon a user configurable event (see objects 2120h – 2125h, 2130h – 2133h, 2140h – 2147h and 2150h – 2153h), can be transmitted upon a SYNC message or when an RTR is received if the sub-indices of this object are configured appropriately. See object 1A14h for details about the data transmitted by this PDO.

| 1814.01h | COB-ID Used By PDO | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | No |
| **Description:**<br>Holds the COB-ID of the PDO as well as other parameters. For details see Section 4.9.4.1. | | | | |

| 1814.02h | Transmission Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 – 255 | N/A | Read / Write | No |
| **Description:**<br>Defines the way in which the PDO will be transmitted, namely synchronous or asynchronous. For details see Section 4.9.4.2. | | | | |

### 1A14h: 21st Transmit PDO Mapping Parameter

This PDO transmits the actual position value stored in the drive. The object mapped to this PDO is fixed and not user selectable. See object 1814h for details on the transmission method.

| 1A14.01h | PDO Mapping for the 1st Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the Actual Position Value object (6064h). For details about the format of this sub-index see Section 4.9.5. | | | | |

### 1815h: 22nd Transmit PDO Communication Parameter

This PDO is transmitted upon a user configurable event (see objects 2120h – 2125h, 2130h – 2133h, 2140h – 2147h and 2150h – 2153h), can be transmitted upon a SYNC message or when an RTR is received if the sub-indices of this object are configured appropriately. See object 1A15h for details about the data transmitted by this PDO.

| 1815.01h | COB-ID Used By PDO | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | No |
| **Description:**<br>Holds the COB-ID of the PDO as well as other parameters. For details see Section 4.9.4.1. | | | | |

| 1815.02h | Transmission Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 – 255 | N/A | Read / Write | No |
| **Description:**<br>Defines the way in which the PDO will be transmitted, namely synchronous or asynchronous. For details see Section 4.9.4.2. | | | | |

### 1A15h: 22nd Transmit PDO Mapping Parameter

This PDO transmits the actual velocity value stored in the drive. The object mapped to this PDO is fixed and not user selectable. See object 1815h for details on the transmission method.

| 1A15.01h | PDO Mapping for the 1st Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the Actual Velocity Value object (606Ch). For details about the format of this sub-index see Section 4.9.5. | | | | |

### 1816h: 23rd Transmit PDO Communication Parameter

This PDO is transmitted upon a user configurable event (see objects 2120h – 2125h, 2130h – 2133h, 2140h – 2147h and 2150h – 2153h), can be transmitted upon a SYNC message or when an RTR is received if the sub-indices of this object are configured appropriately. See object 1A16h for details about the data transmitted by this PDO.

| 1816.01h | COB-ID Used By PDO | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | No |
| **Description:**<br>Holds the COB-ID of the PDO as well as other parameters. For details see Section 4.9.4.1. | | | | |

| 1816.02h | Transmission Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 – 255 | N/A | Read / Write | No |
| **Description:**<br>Defines the way in which the PDO will be transmitted, namely synchronous or asynchronous. For details see Section 4.9.4.2. | | | | |

### 1A16h: 23rd Transmit PDO Mapping Parameter

This PDO transmits the actual torque value stored in the drive. The object mapped to this PDO is fixed and not user selectable. See object 1816h for details on the transmission method.

| 1A16.01h | PDO Mapping for the 1st Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the Actual Current Value object (6077h). For details about the format of this sub-index see Section 4.9.5. | | | | |

### 1817h: 24th Transmit PDO Communication Parameter

This PDO is applicable to interpolated position mode only (see object 6060h for operating modes) and is transmitted upon a user configurable event (see objects 2120h – 2125h, 2130h – 2133h, 2140h – 2147h and 2150h – 2153h). The PDO can also be transmitted upon a SYNC message or when an RTR is received if the sub-indices of this object are configured appropriately. See object 1A17h for details about the data transmitted by this PDO.

| 1817.01h | COB-ID Used By PDO | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | No |

| Description: |
|---|
| Holds the COB-ID of the PDO as well as other parameters. For details see Section 4.9.4.1. |

| 1817.02h | Transmission Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 – 255 | N/A | Read / Write | No |
| **Description:** Defines the way in which the PDO will be transmitted, namely synchronous or asynchronous. For details see Section 4.9.4.2. | | | | |

### 1A17h: 24th Transmit PDO Mapping Parameter

This PDO transmits information about the status of the PVT buffer in the drive. The PDO will only transmit if the drive is in interpolated position mode (see object 6060h for operating modes). The object mapped to this PDO is fixed and not user selectable. See object 1817h for details on the transmission method.

| 1A17.01h | PDO Mapping for the 1st Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:** Maps the Buffer Position of the Interpolation Data Configuration object (60C4.04h). For details about the format of this sub-index see Section 4.9.5. | | | | |

### 1818h: 25th Transmit PDO Communication Parameter

This PDO is applicable to all operating modes (see object 6060h for operating modes) and is transmitted upon an internal drive event. A change in the state of a digital input is the internal event that triggers this PDO. The PDO can also be transmitted upon a SYNC message or when an RTR is received if the sub-indices of this object are configured appropriately. See object 1A18h for details about the data transmitted by this PDO.

| 1818.01h | COB-ID Used By PDO | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | No |
| **Description:** Holds the COB-ID of the PDO as well as other parameters. For details see Section 4.9.4.1. | | | | |

| 1818.02h | Transmission Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 – 255 | N/A | Read / Write | No |
| **Description:** Defines the way in which the PDO will be transmitted, namely synchronous or asynchronous. For details see Section 4.9.4.2. | | | | |

### 1A18h: 25th Transmit PDO Mapping Parameter

This PDO transmits information about the status of the programmable and dedicated digital inputs on the drive. The objects mapped to this PDO are fixed and not user selectable. See object 1818h for details on the transmission method.

| 1A18.01h | PDO Mapping for the 1st Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |

| | | | | |
|---|---|---|---|---|
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the Programmable Digital Inputs object (20A0h). For details about the format of this sub-index see Section 4.9.5. | | | | |

| 1A18.02h | PDO Mapping for the 2nd Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read Only | No |
| **Description:**<br>Maps the Dedicated Digital Inputs object (2004h). For details about the format of this sub-index see Section 4.9.5. | | | | |

### 1819h: 26th Transmit PDO Communication Parameter

This PDO is applicable to all operating modes (see object 6060h for operating modes) and is transmitted upon a user configurable event (see objects 2120h – 2125h, 2130h – 2133h, 2140h – 2147h and 2150h – 2153h). The PDO can also be transmitted upon a SYNC message or when an RTR is received if the sub-indices of this object are configured appropriately. See object 1A19h for details about the data transmitted by this PDO.

| 1819.01h | COB-ID Used By PDO | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | No |
| **Description:**<br>Holds the COB-ID of the PDO as well as other parameters. For details see Section 4.9.4.1. | | | | |

| 1819.02h | Transmission Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 – 255 | N/A | Read / Write | No |
| **Description:**<br>Defines the way in which the PDO will be transmitted, namely synchronous or asynchronous. For details see Section 4.9.4.2. | | | | |

### 1A19h: 26th Transmit PDO Mapping Parameter

This PDO transmits up to 8 user specified objects defined by the sub-indices below. Any object in this object dictionary may be mapped to one of these sub-indices; there is no restriction other than data size. If a large object, such as a 32-byte string, is mapped to TDDO26, it simply will not transmit when triggered. Generally it is most useful to map numerical data to this TPDO.

The total number of bytes TPDO26 can transmit is 8. If, across all the sub-indices, more than 8 bytes are assigned to transmit, TPDO26 will not transmit.

Example 1: Map 8 objects to all 8 sub-indices of TPDO26. Each object only has 8 bits of data, therefore the total bytes to transmit = 8. In this case TPDO26 will transmit and the data will appear sub-index 1 = byte 1, sub-index 2 = byte 2 and so on.

Example 2: Map 2 objects, each a 32-bit object, to sub-indices 1 and 2. In this case TPDO26 will transmit and the data will appear sub-index 1 = bytes 1-4, sub-index 2 = bytes 5-8.

Example 3: Map 3 objects, two 32-bit objects and one 16-bit on object to sub-indices 1, 2, and 3. In this case TPDO26 will not transmit because the total number of bytes assigned to transmit exceeds 8.

See object 1819h for details on setting the transmission method.

| 1A19.01h | PDO Mapping for the 1st Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | No |

**Description:**
Identifies an object that TPDO26 will transmit data from when triggered. It is important to note that TPDO26 only has 8 available data bytes to transmit information with. If sub-indices 1A19.01 through 1A19.08 contain objects such that the total number of bytes to transmit is greater than 8, TPDO26 will not transmit any data.

Ex. Objects 6064 and 606B are mapped to 1A19.01 and 1A19.02. These two objects are of the Unsigned32 data type therefore TPDO26 is configured to transmit a total of 8 bytes when triggered. In this case all zeros should be written to 1A19.03 - 1A19.08 because if there are any other objects mapped to TPDO26, it will not transmit when requested as there are more than 8 bytes assigned to it.

For details about formatting data for this sub-index see Section 4.9.5.

| 1A19.02h | PDO Mapping for the 2nd Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | No |

**Description:**
Identifies an object that TPDO26 will transmit data from when triggered. It is important to note that TPDO26 only has 8 available data bytes to transmit information with. If sub-indices 1A19.01 through 1A19.08 contain objects such that the total number of bytes to transmit is greater than 8, TPDO26 will not transmit any data.

Ex. Objects 6064 and 606B are mapped to 1A19.01 and 1A19.02. These two objects are of the Unsigned32 data type therefore TPDO26 is configured to transmit a total of 8 bytes when triggered. In this case all zeros should be written to 1A19.03 - 1A19.08 because if there are any other objects mapped to TPDO26, it will not transmit when requested as there are more than 8 bytes assigned to it.

For details about formatting data for this sub-index see Section 4.9.5.

| 1A19.03h | PDO Mapping for the 3rd Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | No |

**Description:**
Identifies an object that TPDO26 will transmit data from when triggered. It is important to note that TPDO26 only has 8 available data bytes to transmit information with. If sub-indices 1A19.01 through 1A19.08 contain objects such that the total number of bytes to transmit is greater than 8, TPDO26 will not transmit any data.

Ex. Objects 6064 and 606B are mapped to 1A19.01 and 1A19.02. These two objects are of the Unsigned32 data type therefore TPDO26 is configured to transmit a total of 8 bytes when triggered. In this case all zeros should be written to 1A19.03 - 1A19.08 because if there are any other objects mapped to TPDO26, it will not transmit when requested as there are more than 8 bytes assigned to it.

For details about formatting data for this sub-index see Section 4.9.5.

| 1A19.04h | PDO Mapping for the 4th Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | No |

**Description:**
Identifies an object that TPDO26 will transmit data from when triggered. It is important to note that TPDO26 only has 8 available data bytes to transmit information with. If sub-indices 1A19.01 through 1A19.08 contain objects such that the total number of bytes to transmit is greater than 8, TPDO26 will not transmit any data.

Ex. Objects 6064 and 606B are mapped to 1A19.01 and 1A19.02. These two objects are of the Unsigned32 data type therefore TPDO26 is configured to transmit a total of 8 bytes when triggered. In this case all zeros should be written to 1A19.03 - 1A19.08 because if there are any other objects mapped to TPDO26, it will not transmit when requested as there are more than 8 bytes assigned to it.

For details about formatting data for this sub-index see Section 4.9.5.

| 1A19.05h | PDO Mapping for the 6th Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | No |

**Description:**
Identifies an object that TPDO26 will transmit data from when triggered. It is important to note that TPDO26 only has 8 available data bytes to transmit information with. If sub-indices 1A19.01 through 1A19.08 contain objects such that the total number of bytes to transmit is greater than 8, TPDO26 will not transmit any data.

Ex. Objects 6064 and 606B are mapped to 1A19.01 and 1A19.02. These two objects are of the Unsigned32 data type therefore TPDO26 is configured to transmit a total of 8 bytes when triggered. In this case all zeros should be written to 1A19.03 - 1A19.08 because if there are any other objects mapped to TPDO26, it will not transmit when requested as there are more than 8 bytes assigned to it.

For details about formatting data for this sub-index see Section 4.9.5.

| 1A19.06h | PDO Mapping for the 6th Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | No |

**Description:**
Identifies an object that TPDO26 will transmit data from when triggered. It is important to note that TPDO26 only has 8 available data bytes to transmit information with. If sub-indices 1A19.01 through 1A19.08 contain objects such that the total number of bytes to transmit is greater than 8, TPDO26 will not transmit any data.

Ex. Objects 6064 and 606B are mapped to 1A19.01 and 1A19.02. These two objects are of the Unsigned32 data type therefore TPDO26 is configured to transmit a total of 8 bytes when triggered. In this case all zeros should be written to 1A19.03 - 1A19.08 because if there are any other objects mapped to TPDO26, it will not transmit when requested as there are more than 8 bytes assigned to it.

For details about formatting data for this sub-index see Section 4.9.5.

| 1A19.07h | PDO Mapping for the 7th Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | No |

**Description:**
Identifies an object that TPDO26 will transmit data from when triggered. It is important to note that TPDO26 only has 8 available data bytes to transmit information with. If sub-indices 1A19.01 through 1A19.08 contain objects such that the total number of bytes to transmit is greater than 8, TPDO26 will not transmit any data.

Ex. Objects 6064 and 606B are mapped to 1A19.01 and 1A19.02. These two objects are of the Unsigned32 data type therefore TPDO26 is configured to transmit a total of 8 bytes when triggered. In this case all zeros should be written to 1A19.03 - 1A19.08 because if there are any other objects mapped to TPDO26, it will not transmit when requested as there are more than 8 bytes assigned to it.

For details about formatting data for this sub-index see Section 4.9.5.

| 1A19.08h | PDO Mapping for the 8th Application Object | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | N/A | N/A | Read / Write | No |

**Description:**
Identifies an object that TPDO26 will transmit data from when triggered. It is important to note that TPDO26 only has 8 available data bytes to transmit information with. If sub-indices 1A19.01 through 1A19.08 contain objects such that the total number of bytes to transmit is greater than 8, TPDO26 will not transmit any data.

Ex. Objects 6064 and 606B are mapped to 1A19.01 and 1A19.02. These two objects are of the Unsigned32 data type therefore TPDO26 is configured to transmit a total of 8 bytes when triggered. In this case all zeros should be written to 1A19.03 - 1A19.08 because if there are any other objects mapped to TPDO26, it will not transmit when requested as there are more than 8 bytes assigned to it.

For details about formatting data for this sub-index see Section 4.9.5.

### 2100h: Stored CANbus Baud Rate

| 2100h | Stored CANbus Baud Rate | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | See below* | Kbits/s | Read / Write | Yes |

**Description:**
If the hardware addressing is not used, the drive will default to communicating over the CANbus via the baud rate stored in this object. The default rate is 3E8h (1000) If an invalid number is entered into this object, the drive will reset this value back to the default.
* The drive will accept these valid baud rates: 1000, 500, 250, 125.

### 2111h: SDO Size Indication

| 2111h | SDO Size Indication | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | N/A | N/A | Read / Write | Yes |

**Description:**
This object determines if size indications will be used during SDO messaging. See table below for appropriate values and their effects on the drive.

| Value | Meaning |
|---|---|
| 0 | Drive does not respond with size indications in SDO messages |
| Anything non-Zero | Drive responds with size indications |

### 2120h: TPDO Timer1 Cycle Time

| 2120h | TPDO Timer1 Cycle Time | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | $0 – 2^{32}$ | Milliseconds | Read / Write | Yes |

**Description:**
Sets the cycle time of the assigned TPDOs (assigned in object 2121h). If the cycle time is set to 0, the assigned TPDOs will be transmitted continuously.

### 2121h: TPDO Timer1 Assigned TPDOs

| 2121h | TPDO Timer1 Assigned TPDOs | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | 0 – 1FFh | N/A | Read / Write | Yes |

**Description:**
Assigns TPDOs to Timer1. If this object is set to 0, Timer1 will stop.

| Bit | Assignment (1 = assigned, 0 = not assigned) |
|---|---|
| 0 | TPDO 1 |
| 1 | TPDO 3 |
| 2 | TPDO 4 |
| 3 | TPDO 5 |
| 4 | TPDO 21 |
| 5 | TPDO 22 |
| 6 | TPDO 23 |
| 7 | TPDO 24 |
| 8 | TPDO 25 |
| 9 | TPDO 26 |
| 10 - 31 | Reserved |

### 2122h: TPDO Timer1 Next Processing Time

| 2122h | TPDO Timer1 Next Processing Time | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |

| | | | | |
|---|---|---|---|---|
| Unsigned32 | 0 – $2^{32}$ | Milliseconds | Read | No |

**Description:**
Contains the time of the next Timer1 event with respect to the total drive run time as seen by the drive.

### 2123h: TPDO Timer2 Cycle Time

| 2123h | TPDO Timer2 Assigned TPDOs | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | 0 – $2^{32}$ | Milliseconds | Read / Write | Yes |

**Description:**
Sets the cycle time of the assigned TPDOs for Timer2. If the cycle time is set to 0, the assigned TPDOs will be transmitted continuously.

### 2124h: TPDO Timer2 Assigned TPDOs

| 2124h | TPDO Timer2 Assigned TPDOs | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | 0 – 1FFh | N/A | Read / Write | Yes |

**Description:**
Assigns TPDOs to Timer 1. If this object is set to 0, Timer 1 will stop.

| Bit | Assignment (1 = assigned, 0 = not assigned) |
|---|---|
| 0 | TPDO 1 |
| 1 | TPDO 3 |
| 2 | TPDO 4 |
| 3 | TPDO 5 |
| 4 | TPDO 21 |
| 5 | TPDO 22 |
| 6 | TPDO 23 |
| 7 | TPDO 24 |
| 8 | TPDO 25 |
| 9 | TPDO 26 |
| 10 - 31 | Reserved |

### 2125h: TPDO Timer2 Next Processing Time

| 2125h | TPDO Timer2 Next Processing Time | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | 0 – $2^{32}$ | Milliseconds | Read | No |

**Description:**
Contains the time of the next Timer2 event with respect to the total drive run time as seen by the drive.

### 2130h: TPDO Value-Changed Object ID

| 2130h | TPDO Value-Changed Object ID | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | $0 – 2^{32}$ | N/A | Read / Write | No |

**Description:**
Contains the Object ID of the object to observe continuously. After a user specified value change of this object (set via object 2131h), the assigned TPDOs will be sent (assigned via object 2132h). Use the three objects (2130h, 2131h, 2132h) to monitor any object and send assigned TPDOs after a desired value change. Use the format in the table below to specify the observed object.

| Byte0 | Byte1 | Byte2 | Byte3 |
|---|---|---|---|
| Sub-index | Object Index (LSB) | Object Index (MSB) | Always 0 |

### 2131h: TPDO Value-Changed Delta Value

| 2131h | TPDO Value-Changed Delta Value | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | $0 – 2^{32}$ | N/A | Read / Write | No |

**Description:**
Sets the amount of change of the observed object (defined by 2130h) that will cause the assigned Transmit PDOs to be sent (assigned via object 2132h). Use the three objects (2130h, 2131h, 2132h) to monitor any object and send assigned TPDOs after a desired value change. Setting this value to zero disables the functionality. The meaning of the value in this object depends on the observed object.

### 2132h: TPDO Value-Changed Assigned TPDOs

| 2132h | TPDO Value-Changed Assigned TPDOs | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | 0 – 1FFh | N/A | Read / Write | Yes |

**Description:**
Assigns TPDOs to Value-Changed event. If this object is set to 0, Timer 1 will stop.

| Bit | Assignment (1 = assigned, 0 = not assigned) |
|---|---|
| 0 | TPDO 1 |
| 1 | TPDO 3 |
| 2 | TPDO 4 |
| 3 | TPDO 5 |
| 4 | TPDO 21 |
| 5 | TPDO 22 |
| 6 | TPDO 23 |
| 7 | TPDO 24 |
| 8 | TPDO 25 |
| 9 | TPDO 26 |
| 10 - 31 | Reserved |

### 2133h: TPDO Value-Changed Object Last Value

| 2133h | TPDO Value-Changed Object Last Value | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | $0 – 2^{32}$ | N/A | Read Only | No |

**Description:**
This object consists of the value of the observed object, defined by 2130h, from the last TPDO transmission triggered by a Value-Changed event.

### 2140h: TPDO Bits-Changed_1 Object ID

| 2140h | TPDO Bits-Changed_1 Object ID | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | $0 – 2^{32}$ | N/A | Read / Write | Yes |

**Description:**
This object is used to identify a CANopen object which is observed continuously for bit changing. If the observed bits change, the assigned TPDOs will be sent. The observed bits are defined by a bit mask in object 2141h while the assigned TPDOs are defined by object 2142h. Use the format in the table below to specify the observed object.

| Byte0 | Byte1 | Byte2 | Byte3 |
|---|---|---|---|
| Sub-index | Object Index (LSB) | Object Index (MSB) | Always 0 |

### 2141h: TPDO Bits-Changed_1 Object Bit Mask

| 2141h | TPDO Bits-Changed_1 Object Bit Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | $0 – 2^{32}$ | N/A | Read / Write | Yes |

**Description:**
This object consists of a bit mask to identify which bits are observed in the object identified in 2140h . If the observed bits change the assigned TPDOs are sent. If this variable is set to 0 the identified object will not be observed.

### 2142h: TPDO Bits-Changed_1 Assigned TPDOs

| 2142h | TPDO Bits-Changed_1 Assigned TPDOs | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | $0 – 2^{32}$ | N/A | Read / Write | Yes |

**Description:**
Assigns TPDOs to Bits-Changed1 event. If this object is set to a value of 0, the object identified in 2140h will not be observed.

| Bit | Assignment (1 = assigned, 0 = not assigned) |
|---|---|
| 0 | TPDO 1 |
| 1 | TPDO 3 |
| 2 | TPDO 4 |
| 3 | TPDO 5 |
| 4 | TPDO 21 |
| 5 | TPDO 22 |
| 6 | TPDO 23 |
| 7 | TPDO 24 |
| 8 | TPDO 25 |
| 9 | TPDO 26 |
| 10 - 31 | Reserved |

### 2143h: TPDO Bits-Changed_1 Object Last Value

| 2143h | TPDO Bits-Changed_1 Object Last Value | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | $0 – 2^{32}$ | N/A | Read Only | No |

**Description:**
This object consists of the value of the observed object, defined by 2140h, from the last TPDO transmission triggered by a Bits-Changed1 event.

### 2144h: TPDO Bits-Changed_2 Object ID

| 2144h | TPDO Bits-Changed_2 Object ID | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | $0 – 2^{32}$ | N/A | Read / Write | Yes |

**Description:**
This object is used to identify a CANopen object which is observed continuously for bit changing. If the observed bits change, the assigned TPDOs will be sent. The observed bits are defined by a bit mask in object 2145h while the assigned TPDOs are defined by object 2146h. Use the format in the table below to specify the observed object.

### 2145h: TPDO Bits-Changed_2 Object Bit Mask

| 2145h | TPDO Bits-Changed_2 Object Bit Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | $0 – 2^{32}$ | N/A | Read / Write | Yes |

**Description:**
This object consists of a bit mask to identify which bits are observed in the object identified in 2140h . If the observed bits change the assigned TPDOs are sent. If this variable is set to 0 the identified object will not be observed.

### 2146h: TPDO Bits-Changed_2 Assigned TPDOs

| 2146h | TPDO Bits-Changed_2 Assigned TPDOs | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | $0 – 2^{32}$ | N/A | Read / Write | Yes |

**Description:**
Assigns TPDOs to Bits-Changed2 event. If this object is set to a value of 0, the object identified in 2144h will not be observed.

| Bit | Assignment (1 = assigned, 0 = not assigned) |
|---|---|
| 0 | TPDO 1 |
| 1 | TPDO 3 |
| 2 | TPDO 4 |
| 3 | TPDO 5 |
| 4 | TPDO 21 |
| 5 | TPDO 22 |
| 6 | TPDO 23 |
| 7 | TPDO 24 |
| 8 | TPDO 25 |
| 9 | TPDO 26 |
| 10 - 31 | Reserved |

### 2147h: TPDO Bits-Changed_2 Object Last Value

| 2147h | TPDO Bits-Changed_2 Object Last Value | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | $0 – 2^{32}$ | N/A | Read Only | No |

**Description:**
This object consists of the value of the observed object, defined by 2144h, from the last TPDO transmission triggered by a Bits-Changed2 event.

### 2150h: TPDO Value-Reached Object ID

| 2150h | TPDO Value-Reached Object ID | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | $0 – 2^{32}$ | N/A | Read / Write | Yes |

**Description:**
This object is used to identify a CANopen object which is observed continuously for changing. If the value of the observed object reaches a predefined value, the assigned TPDOs will be sent. The predefined value is defined in 2151h while the assigned TPDOs are defined in 2152h. Use the format in the table below to specify the observed object.

| Byte0 | Byte1 | Byte2 | Byte3 |
|---|---|---|---|
| Sub-index | Object Index (LSB) | Object Index (MSB) | Always 0 |

### 2151h: TPDO Value-Reached

| 2151h | TPDO Value-Reached | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | $0 – 2^{32}$ | N/A | Read / Write | Yes |

**Description:**
This object consists of a predefined value to compare with the value of an observed object identified in 2150h . If the value of the observed object reaches this value the assigned TPDOs are sent.

### 2152h: TPDO Value-Reached Assigned TPDOs

| 2152h | TPDO Value-Reached Assigned TPDOs | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | $0 – 2^{32}$ | N/A | Read / Write | Yes |

**Description:**
Assigns TPDOs to Value-Reached event. If this object is set to a value of 0, the object identified in 2150h will not be observed.

| Bit | Assignment (1 = assigned, 0 = not assigned) |
|---|---|
| 0 | TPDO 1 |
| 1 | TPDO 3 |
| 2 | TPDO 4 |
| 3 | TPDO 5 |
| 4 | TPDO 21 |
| 5 | TPDO 22 |
| 6 | TPDO 23 |
| 7 | TPDO 24 |
| 8 | TPDO 25 |
| 9 | TPDO 26 |
| 10 - 31 | Reserved |

### 2153h: TPDO Value-Reached Direction

| 2153h | TPDO Value-Reached Direction | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |

| | | | | |
|---|---|---|---|---|
| Unsigned16 | $0 – 2^{16}$ | N/A | Read / Write | Yes |

**Description:**
If the value of this object is 0, the assigned TPDOs (defined by 2152h) are sent if the observed object (identified in 2150h) reaches the predefined value (set by 2151h) in the downward direction. Otherwise the assigned TPDOs are sent if the value of the observed object reaches the predefined value in the upward direction.

### 2009h: Restore Drive Parameters

| 2009.01h | Restore drive parameters key | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | See Table | N/A | Write Only | No |

**Description:**

This object defines which parameters will be restored from the drives non-volatile memory to the current project file.

| Key (Hex) | Description |
|---|---|
| 165B | Restore CANopen communication parameters |
| 1CAE | Restore RS232 communication parameters |
| 7405 | Restore non-axis parameters |
| 8137 | Restore axis parameters |

### 200Ah: Store Drive Parameters

| 200A.01h | Store drive parameters key | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | See Table | N/A | Write Only | Yes |

**Description:**

This object defines which parameters will be stored to the drive's non-volatile memory.

| Key (Hex) | Description |
|---|---|
| 1CAE | Store CANopen communication parameters |
| 165B | Store RS232 communication parameters |
| 7405 | Store non-axis parameters |
| 8137 | Store axis parameters |

### 205Bh: Programmable Status Parameters

This object determines which events will be mapped to the Statusword (6041h) bits, indicated below. When multiple events are mapped to a single bit, they will be logically OR-ed.

Table 23: Object 205B Mapping

| Programmable Status Mask | Description |
|---|---|
| Bit 0…12 | Reserved |
| Bit 13 | Bit 11 (Internal Limit Active) in 6041h (Statusword) |
| Bit 14 | Bit 7 (Warning) in 6041h (Statusword) |
| Bit 15 | Bit 8 (manufacturer specific) in 6041h (Statusword) |

## 208Ch: Product Information

| 208C.01h | Hardware Information | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| String(352) | ASCII | N/A | Read Only | Yes |

**Description:**
Object 208C.01 provides all the drive information in a single 352-byte string. The meaning of each byte in the string is divided into sections according to the following table. Bytes 2 through 33 provide the “Control Board Name” for example.
Contrary to the way most string objects work, these objects are organized Most-Significant-Byte-First when reading them from the drive. Use the Segmented Read Protocol described in SDO Messages (section 4.6.1) of this manual to read 55 seven-byte data packages from this object. Every byte after number 352 should be ignored.

| Byte Definitions | Description |
|---|---|
| 0…1 | Reserved |
| 2…33 | Control Board Name |
| 34…65 | Control Board Version |
| 66…97 | Control Board Serial Number |
| 98…129 | Control Board Build Date |
| 130…161 | Control Board Build Time |
| 162…191 | Reserved |
| 192…223 | Product Part Number (including revision letter) |
| 224…255 | Product Version |
| 256…287 | Product Serial Number |
| 288…319 | Product Build Date |
| 320…352 | Product Build Time |

## 208Dh: Firmware Information

| 208D.01h | Firmware Version | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| String(32) | ASCII | N/A | Read Only | Yes |

**Description:**
This object returns a 32-byte string containing the firmware version that is currently running on this drive. Use the Segmented Read Protocol described in SDO Messages (section 4.6.1) of this manual to read 5 seven-byte data packages from this object. Each 7 byte package is arranged Least Significant Byte first. To arrange the data into readable text, re-organize each package MSB first and place the first package on the far left and the 5th package on the far right.
Example:
Package1: 30 30 33 72 63 64 5A
Package2: 2E 35 2E 32 20 65 65
Package3: 00 00 00 00 00 00 31
Package4: 00 00 00 00 00 00 00
Package5: 00 00 00 00 00 00 00
Arrange to:
5A 64 63 72 33 30 30  65 65 20 32 2E 35 2E  31 00 00 00 00 00 00  00 00 00 00 00 00 00 = Zdcr300ee2.5.1

| 208D.02h | Bootloader Version | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| String(32) | ASCII | N/A | Read Only | Yes |

**Description:**
This object returns a 32-byte string containing the bootloader version that is currently running on this drive. Use the Segmented Read Protocol described in SDO Messages (section 4.6.1) of this manual to read 5 seven-byte data packages from this object. Each 7 byte package is arranged Least Significant Byte first. To arrange the data into readable text, re-organize each package MSB first and place the first package on the far left and the 5th package on the far right.
Example:
Package1: 30 30 33 72 63 64 5A
Package2: 2E 35 2E 32 20 65 65
Package3: 00 00 00 00 00 00 31
Package4: 00 00 00 00 00 00 00
Package5: 00 00 00 00 00 00 00
Arrange to:
5A 64 63 72 33 30 30  65 65 20 32 2E 35 2E  31 00 00 00 00 00 00  00 00 00 00 00 00 00 = Zdcr300ee2.5.1

| 208D.03h | FPGA-Image Version | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| String(32) | ASCII | N/A | Read Only | Yes |

**Description:**
This object returns a 32-byte string containing the FPGA-image version that is currently running on this drive. Use the Segmented Read Protocol described in SDO Messages (section 4.6.1) of this manual to read 5 seven-byte data packages from this object. Each 7 byte package is arranged Least Significant Byte first. To arrange the data into readable text, re-organize each package MSB first and place the first package on the far left and the 5th package on the far right.
Example:
Package1: 30 30 33 72 63 64 5A
Package2: 2E 35 2E 32 20 65 65
Package3: 00 00 00 00 00 00 31
Package4: 00 00 00 00 00 00 00
Package5: 00 00 00 00 00 00 00
Arrange to:
5A 64 63 72 33 30 30  65 65 20 32 2E 35 2E  31 00 00 00 00 00 00  00 00 00 00 00 00 00 = Zdcr300ee2.5.1

## 200Bh: Stored User Parameters

| 200B.01h | User defined drive name | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| String256 | ASCII Values | N/A | Read / Write | Yes |

**Description:**
This object contains a user specified drive name for the drive. The characters in the string are stored as ASCII values. For the drive name "AMC", the digits stored are:

41h, 4Dh, 43h

## 2008h: Drive Initialization Parameters

| 2008D.01h | Start-up sequence control | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | $0 – [2^{(16)} – 1]$ | N/A | Read / Write | Yes |

**Description:**
This object defines how the drive will behave when power is first applied.

| Bit | Drive Initialization Parameters |
|---|---|
| 0 | Inhibit Bridge |
| 1 | Enable Brake |
| 2 | Phase Detect |
| 3 | Load Measured Position |
| 4 | Load Target |
| 5…15 | Reserved |

## 2032h: Feedback Sensor Parameters

| 2032.01h | Encoder wiring polarity | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | $0 – [2^{(16)} – 1]$ | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to the encoder wiring polarity

| 2032.02h | Maximum phase detection current | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $0 – [2^{(31)}-1]$ | DCU | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the maximum phase detection current that is allowed during a phase detect. See Appendix A for DCU (Drive Current Units).

| 2032.03h | Maximum phase detection time limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | 0 – [$2^{(32)}$ –1] | N/A | Read / Write | Yes |
| **Description:**<br><br>This object contains a value corresponding to the maximum phase detection time limit | | | | |

| 2032.04h | Maximum phase detection brake time | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | 0 – [$2^{(32)}$ –1] | N/A | Read / Write | Yes |
| **Description:**<br><br>This object contains a value corresponding to the maximum phase detection brake time | | | | |

| 2032.05h | Maximum phase detection motion | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | DPU | Read / Write | Yes |
| **Description:**<br>This object contains a value corresponding to the maximum phase detection motion that is allowed during a phase detect. See Appendix A for DPU (Drive Position Units). | | | | |

| 2032.06h | Resolver resolution | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |
| **Description:**<br><br>This object contains a value corresponding to the resolver resolution | | | | |

| 2032.07h | Serial encoder type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |
| **Description:**<br><br>This object contains a value corresponding to the serial encoder type | | | | |

| 2032.08h | Position interpolation | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |
| **Description:**<br><br>This object contains a value corresponding to the position interpolation | | | | |

| 2032.09h | Encoder steps per encoder sine period | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

| Description: |
|---|
| This object contains a value corresponding to the encoder steps per encoder sine period |

| 2032.0Ah | Secondary encoder position interpolation | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |
| **Description:** | | | | |
| This object contains a value corresponding to the secondary encoder position interpolation | | | | |

| 2032.0Bh | Torque constant over Inertia | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | [$-2^{(31)}$] – [$2^{(31)}$-1] | N/A | Read / Write | Yes |
| **Description:** | | | | |
| This object contains a value corresponding to the torque constant over inertia. | | | | |

## 2033h: User Voltage Protection Parameters

| 2033.01h | Over-voltage limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | [$-2^{(15)}$] – [$2^{(15)}$-1] | DVU | Read Only | No |
| **Description:** | | | | |
| This object contains the over voltage limit specified for the drive. It must be set lower than the drive over-voltage hardware shutdown point and greater than the Nominal DC Bus Voltage. See Appendix A for DVU (Drive Voltage Units). | | | | |

| 2033.02h | Under-voltage limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | [$-2^{(15)}$] – [$2^{(15)}$-1] | DVU | Read Only | No |
| **Description:** | | | | |
| This object contains the under voltage limit specified for the drive. It must be set above the drive under-voltage hardware shutdown point and less than the Nominal DC Bus Voltage. See Appendix A for DVU (Drive Voltage Units). | | | | |

| 2033.03h | Shunt regulator enable threshold | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | 0 – [$2^{(15)}$-1] | DVU | Read Only | No |

**Description:**
This object contains a value corresponding to the shunt regulator enable threshold voltage. When the bus reaches this voltage, built in shut regulator will turn on allow excess energy to be dissipated across an external shunt resistor. Not all drives have built in shunt regulators.

See Appendix A for DVU (Drive Voltage Units).

Note: This object does not enable the shunt regulator. To enable the regulator, program this object with the required shunt regulator enable threshold voltage, then use object 2033.04h to enable the shunt.

| 2033.04h | Shunt regulator configuration | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | See Table | N/A | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the current state of the shunt regulator

| Value (Hex) | Description |
|---|---|
| 00 | Shunt Regulator is **OFF** |
| 02 | Shunt Regulator is **ON** |

| 2033.05h | External shunt resistance | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | $0 – [2^{(16)} – 1]$ | DRU | Read / Write | Yes |

**Description:**

This object contains a value corresponding to the resistance of the external shunt resistor. See Appendix A for DRU (Drive Resistance Units).

| 2033.06h | External shunt power | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | $0 – [2^{(16)} – 1]$ | DWU | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the amount of power the external shunt resistor is allowed to dissipate. See Appendix A for DWU (Drive Power Units).

| 2033.07h | External shunt inductance | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | $0 – [2^{(16)} – 1]$ | DHU | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the inductance of the external shunt resistor. See Appendix A for DHU (Drive Inductance Units).

## 2034h: Current Loop & Commutation Control Parameters

| 2034.01h | Torque current loop proportional gain | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | 0 – [$2^{(15)}$-1] | N/A | Read / Write | Yes |

**Description:**
This object contains the value of proportional gain for the current loop.

$$2^{\frac{\log(Gain)}{\log(2)}+1} X\,256 = \text{Value in 2034.01h} \quad (\text{for } 0 \le \text{Gain} \le 64)$$

To set a Torque Current Loop proportional gain of 40:

$$2^{\frac{\log(40)}{\log(2)}+1} X\,256 = 10240d \; or \; 2800h$$

| 2034.02h | Torque current loop integral gain | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | 0 – [$2^{(15)}$-1] | N/A | Read / Write | Yes |

**Description:**

This object contains the value of integral gain for the current loop.

$$2^{\frac{\log(Gain)}{\log(2)}+1} X\,256 = \text{Value in 2034.01h} \quad (\text{for } 0 \le \text{Gain} \le 64)$$

To set a torque current loop integral gain of 0.75:

$$2^{\frac{\log(0.75)}{\log(2)}+1} X\,256 = 384d \; or \; 0180h$$

| 2034.03h | Torque current target offset | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | [$-2^{(15)}$] – [$2^{(15)}$-1] | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to the torque current target offset

| 2034.04h | Peak current limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | 0 – [$2^{(15)}$-1] | DCU | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the peak current limit set in the drive. See Appendix A for DCU (Drive Current Units).

| 2034.05h | Peak current hold time | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the peak current time set in the drive.

| 2034.06h | Continuous current limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | 0 – [$2^{(15)}$-1] | DCU | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the continuous current limit set in the drive. See Appendix A for DCU (Drive Current Units).

| 2034.07h | Peak to continuous current transition time | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the peak to continuous current transition time set in the drive.

| 2034.08h | Flux current reference loop proportional gain | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | [$-2^{(31)}$] – [$2^{(31)}$-1] | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to the flex current reference loop proportional gain. The flux current loop is only used for AC induction motors

$\frac{\text{Value in 2034.08}}{\text{10000h}}$ =Flux Current Reference Loop Proportional Gain ( $0 \leq \text{Gain} \leq 32767$ )

| 2034.09h | Flux current reference loop integral gain | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | [$-2^{(31)}$] – [$2^{(31)}$-1] | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to the flex current reference loop integral gain. The flux current loop is only used for AC induction motors

$\frac{\text{Value in 2034.09}}{\text{400000h}}$ =Flux Current Reference Loop Integral Gain ( $0 \leq \text{Gain} \leq 512$)

| 2034.0Ah | Motor pole pairs | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to the number of motor pole pairs that are present when an AC induction motor is used. The object has no effect for other motor types.

| 2034.0Bh | Rated peak line current | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to the rated peak line current allowed when using an AC induction motor.

| 2034.0Ch | No load peak magnetization current | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to the No load peak magnetization current allowed when using an AC induction motor.

| 2034.0Dh | Rated frequency | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to the rated frequency

| 2034.0Eh | Rated rotor no load base speed | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | $0 – [2^{(16)} –1]$ | RPM | Read / Write | Yes |

**Description:**

This object contains a value corresponding to the rated rotor no load base speed. This parameter is only used with an AC induction motor.

| 2034.0Fh | FW threshold speed | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | $0 – [2^{(16)} –1]$ | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to the field weakening threshold speed. This parameter is used for AC induction motors only

| 2034.10h | Motor Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |

| | | | | |
|---|---|---|---|---|
| Unsigned16 | - | N/A | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the type of motor connected to the drive.

| 2034.11h | Auxiliary commutation mode | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | - | N/A | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the auxiliary commutation mode. Auxiliary commutation only occurs if the drive is connected to a ***brushed*** motor. Brushed motors commutate the motor internally and therefore do not require the drive to commutate the motor. The drive supplies current over two phases. This remains fixed for a brushed drive.

| 2034.12h | Encoder Direction | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | - | N/A | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the direction of the encoder feedback.

| 2034.13h | Synchronization mode | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | - | N/A | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the current commutation method.

| 2034.14h | Encoder counts per electrical cycle | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $0 – [2^{(31)}-1]$ | Counts | Read / Write | Yes |

**Description:**
This object contains the number of encoder counts per electrical cycle.

| 2034.15h | NTHS angle 1 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | $0 – [2^{(16)} –1]$ | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to the NTHS angle 1.

| 2034.16h | NTHS angle 2 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | $0 – [2^{(16)} –1]$ | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to the NTHS angle 2.

| 2034.17h | Max SPA adjustment | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |
| **Description:**<br><br>This object contains a value corresponding to the max SPA adjustment. | | | | |

| 2034.18h | NTIS angle 1 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |
| **Description:**<br><br>This object contains a value corresponding to the NTIS angle 1. | | | | |

| 2034.19h | EC adjust count | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |
| **Description:**<br><br>This object contains a value corresponding to the EC adjust count. | | | | |

| 2034.1Ah | NTIS angle 2 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |
| **Description:**<br><br>This object contains a value corresponding to the NTIS angle 2. | | | | |

| 2034.1Bh | ECC adjust amount | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | [$-2^{(15)}$] – [$2^{(15)}$-1] | N/A | Read / Write | Yes |
| **Description:**<br><br>This object contains a value corresponding to the ECC adjust amount. | | | | |

| 2034.1Ch | NTA-EZ position | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |
| **Description:**<br><br>This object contains a value corresponding to the NTA-EZ position. | | | | |

| 2034.1Dh | Valid HS mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to the valid HS mask.

| 2034.1Eh | Max SPA error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to the max SPA error.

| 2034.1Fh | Hall Parameter 1 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to Hall Parameter 1.

| 2034.20h | Hall Parameter 2 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to Hall Parameter 2.

| 2034.21h | Hall Parameter 3 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to Hall Parameter 3.

| 2034.22h | Hall Parameter 4 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to Hall Parameter 4.

| 2034.23h | Hall Parameter 5 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to Hall Parameter 5.

| 2034.24h | Hall Parameter 6 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to Hall Parameter 6.

| 2034.25h | Hall Parameter 7 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to Hall Parameter 7.

| 2034.26h | Hall Parameter 8 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**

This object contains a value corresponding to Hall Parameter 8.

## 2036h: Velocity Loop Control Parameters

| 2036.01h | Velocity feedback direction | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | - | N/A | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the feedback polarity of an auxiliary encoder used for velocity feedback.

| 2036.02h | Velocity feedback filter coefficient | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | 0 – [$2^{(31)}$-1] | N/A | Read / Write | Yes |

**Description:**
This object contains a value that corresponds to the velocity feedback filter coefficient.

| 2036.03h | Velocity loop proportional gain | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | 0 – [$2^{(31)}$-1] | N/A | Read / Write | Yes |

**Description:**
This object contains a value that corresponds to the proportional loop gain of the velocity loop.

$\frac{\text{Value in Object 2036.03 (hex)}}{\text{666h}}$ =Velocity Loop Proportional Gain (0 ≤ Velocity Proportional Gain ≤ 1,310,700d)

| 2036.04h | Velocity loop integral gain | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | 0 – [$2^{(31)}$-1] | N/A | Read / Write | Yes |

**Description:**
This object contains a value that corresponds to the integral loop gain of the velocity loop.

Velocity Loop Integral Gain X 06666666h = Value in Object 2036.04h (1 ≤ Velocity Integral Gain ≤ 20d)

Note: For zero velocity loop integral loop gain, store a 0h in object 2036.04h

| 2036.05h | Velocity loop derivative gain | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | 0 – [$2^{(31)}$-1] | N/A | Read / Write | Yes |

**Description:**
This object contains a value that corresponds to the derivative loop gain of the velocity loop.

Velocity Loop Derivative Gain X 0666h = Value in Object 2036.05h (1 ≤ Velocity Integral Gain ≤ 80d)

Note: For zero velocity loop derivative loop gain, store a 0h in object 2036.05h

| 2036.06h | Velocity loop acceleration feed forward gain | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | 0 – [$2^{(31)}$-1] | N/A | Read / Write | Yes |

**Description:**
This object contains a value that corresponds to the velocity loop acceleration feed forward gain.

Velocity Loop Acceleration Feed Forward Gain X 0666h = Value in Object 2036.06h

Where:

(1 ≤ Velocity Loop Acceleration Feed Forward Gain ≤ 80d)

Note: For zero velocity loop velocity loop acceleration feed forward gain, store a 0h in object 2036.06h

## 2037h: Velocity Limits

| 2037.01h | Motor over speed limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | 0 – [$2^{(31)}$-1] | DSU | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the motor over speed limit set in the drive. When the velocity of the motor meets or exceeds this value, the drive will indicate a motor over speed condition is present. See Appendix A for DSU (Drive Velocity Units).

| 2037.02h | Zero speed limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | 0 – [$2^{(31)}$-1] | DSU | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the motor zero speed limit set in the drive. When the velocity of the motor reaches this value or LOWER, the drive will indicate that it has reached a zero speed condition. See Appendix A for DSU (Drive Velocity Units).

| 2037.03h | Velocity at speed limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | 0 – [$2^{(31)}$-1] | DSU | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the velocity at speed limit set in the drive. When the velocity of the motor reaches this value or LOWER, the drive will indicate that it has reached its target velocity. See Appendix A for DSU (Drive Velocity Units).

| 2037.04h | Velocity loop following error limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | [$-2^{(31)}$] – [$2^{(31)}$-1] | DSU | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the velocity at speed limit set in the drive. If the measured velocity meets or exceeds this value, the drive will perceive this as a velocity following error. See Appendix A for DSU (Drive Velocity Units).

| 2037.05h | Positive velocity limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | 0 – [$2^{(31)}$-1] | DSU | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the positive velocity limit set in the drive. When the speed set by this value is met or exceeded, the drive will indicate that the positive limit was reached. See Appendix A for DSU (Drive Velocity Units).

| 2037.06h | Negative velocity limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | 0 – [$2^{(31)}$-1] | DSU | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the negative velocity limit set in the drive. When the speed set by this value is met or exceeded, the drive will indicate that the negative limit was reached. See Appendix A for DSU (Drive Velocity Units).

## 2038h: Position Loop Control Parameters

| 2038.01h | Position loop proportional gain | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | 0 – [$2^{(31)}$-1] | N/A | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the position loop proportional gain. The value of this object can be calculated using this formula:

$$\frac{\text{Value stored in 2038.01h}}{\text{68DB8h}} = \text{Position Loop Proportional Gain (where } 0 \leq \text{Gain} \leq 5000)$$

| 2038.02h | Position loop integral gain | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | 0 – [$2^{(31)}$-1] | N/A | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the position loop integral gain. The value of this object can be calculated using this formula:

$$\frac{\text{Value stored in 2038.02h}}{\text{68DB8BB1Eh}} = \text{Position Loop Integral Gain (where } 0 \leq \text{Gain} \leq 0.076293363)$$

| 2038.03h | Position loop derivative gain | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | 0 – [$2^{(31)}$-1] | N/A | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the position loop derivative gain. The value of this object can be calculated using this formula:

$$\frac{\text{Value stored in 2038.03h}}{\text{68DBh}} = \text{Position Loop Derivative Gain (where } 0 \leq \text{Gain} \leq 79999.999)$$

| 2038.04h | Position loop velocity feed forward gain | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | 0 – [$2^{(31)}$-1] | N/A | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the position loop velocity feed forward gain. The value of this object can be calculated using this formula:

$$\frac{\text{Value stored in 2038.04h}}{\text{68DBh}} = \text{Position Loop Velocity Feed Forward Gain}$$

(where $0 \leq \text{Gain} \leq 39999.999$)

| 2038.05h | Position loop acceleration feed forward gain | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | 0 – [$2^{(31)}$-1] | N/A | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the position loop acceleration feed forward gain. The value of this object can be calculated using this formula:

$$\frac{\text{Value stored in 2038.05h}}{\text{68DBh}} = \text{Position Loop acceleration Feed Forward Gain}$$

(where $0 \leq \text{Gain} \leq 39999.999$)

| 2038.06h | Position feedback direction | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | - | N/A | Read / Write | Yes |

**Description:**
This object contains a value corresponding to the feedback polarity of an auxiliary encoder used for position feedback.

## 2039h: Position Limits

| 2039.01h | Measured Position Value | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] - [2^{(31)} - 1]$ | DPU | Read / Write | Yes |

**Description:**
Replacement value for the measured position when the Load Measured Position event is triggered. This allows you to redefine the current measured position (e.g. reset to zero). See Appendix A for DPU (Drive Position Units).

| 2039.02h | Home Position Value | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] - [2^{(31)} - 1]$ | DPU | Read / Write | Yes |

**Description:**
Position value of the home position. When the measured position reaches this position, within the In-Home Position Window, the At-Home event becomes active. See Appendix A for DPU (Drive Position Units).

| 2039.03h | Max Measured Position Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] - [2^{(31)} - 1]$ | DPU | Read / Write | Yes |

**Description:**
Maximum allowed measured position. The Max Measured Position event will become active if the measured position exceeds this value. See Appendix A for DPU (Drive Position Units).

| 2039.04h | Min Measured Position Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] - [2^{(31)} - 1]$ | DPU | Read / Write | Yes |

**Description:**
Minimum allowed measured position. The Min Measured Position event will become active if the measured position exceeds this value. See Appendix A for DPU (Drive Position Units).

| 2039.05h | In Home Position Window | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] - [2^{(31)} - 1]$ | DPU | Read / Write | Yes |

**Description:**
Defines a window around the Home Position Value, such that when the measured position is within this window, the At-Home event will be active. See Appendix A for DPU (Drive Position Units).

| 2039.06h | In Position Window | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $0 - [2^{(32)} - 1]$ | DPU | Read / Write | Yes |

**Description:**
Defines a window around the target position, such that when the measured position is within this window, the At Command event will be active. See Appendix A for DPU (Drive Position Units).

| 2039.07h | Position Following Error Window | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $0 - [2^{(32)} - 1]$ | DPU | Read / Write | Yes |

**Description:**
This object is the same as DS401's 6065h "Position Following Error Limit".
The maximum allowed position error (difference between demand position and measured position), prior to setting the "Position Following Error" event (active in position mode only). See Appendix A for DPU (Drive Position Units).

| 2039.08h | Max Target Position Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] - [2^{(31)} - 1]$ | DPU | Read / Write | Yes |

**Description:**

Maximum allowed target position. The Max Target Position event will become active if the target position exceeds this value. See Appendix A for DPU (Drive Position Units).

| 2039.09h | Min Target Position Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] - [2^{(31)} - 1]$ | DPU | Read / Write | Yes |

**Description:**

Minimum allowed target position. The Min Target Position event will become active if the target position exceeds this value. See Appendix A for DPU (Drive Position Units).

| 2039.0Ah | Position Limits Control | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | - | N/A | Read / Write | Yes |
| **Description:**<br>Defines if the position limits are enabled or not. | | | | |

### 203Ah: Homing Configuration Parameters

| 203A.01h | Homing Speed During Search For Switch | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | 0- [$2^{(32)}$ –1] | DSU | Read / Write | Yes |
| **Description:**<br>The magnitude of the velocity to be used during the search for the switch (before searching for the home/zero position). See Appendix A for DSU (Drive Velocity Units). | | | | |

| 203A.02h | Homing Speed During Search For Zero | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | 0- [$2^{(32)}$ –1] | DSU | Read / Write | Yes |
| **Description:**<br>The magnitude of the velocity to be used during the search for the home/zero position. See Appendix A for DSU (Drive Velocity Units). | | | | |

| 203A.03h | Homing Method | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |
| **Description:**<br>The type of homing routine used. See Homing section for routine descriptions. | | | | |

| 203A.04h | Homing Acceleration | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | 0- [$2^{(32)}$ –1] | DSU | Read / Write | Yes |
| **Description:**<br>The acceleration and deceleration used during the search for the switch and during the search for zero. See Appendix A for DSU (Drive Velocity Units). | | | | |

### 203Ch: Command Profiler Parameters

The command profiler limits the slope of the target command in any mode. It is broken into four components, where each component is assigned to one sub-index. To remove any effects of the command profiler, set the value of each sub-index to the maximum.

| 203C.01h | Profiler Positive Target Positive Change | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned48 | 0- [$2^{(48)}$ –1] | DJU, DSU, or DAU | Read / Write | Yes |
| **Description:**<br>Defines the maximum positive change in positive command used with the command profiler. Mode dependant. See Appendix A for unit conversions. | | | | |

| 203C.02h | Profiler Positive Target Negative Change | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned48 | 0- [$2^{(48)}$ –1] | DJU, DSU, or DAU | Read / Write | Yes |
| **Description:**<br>Defines the maximum negative change in positive command used with the command profiler. | | | | |

| 203C.03h | Profiler Negative Target Negative Change | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned48 | 0- [$2^{(48)}$ –1] | DJU, DSU, or DAU | Read / Write | Yes |
| **Description:**<br>Defines the maximum negative change in negative command used with the command profiler. Mode dependant. See Appendix A for unit conversions. | | | | |

| 203C.04h | Profiler Negative Target Positive Change | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned48 | 0- [$2^{(48)}$ –1] | DJU, DSU, or DAU | Read / Write | Yes |
| **Description:**<br>Defines the maximum positive change in negative command used with the command profiler. Mode dependant. See Appendix A for unit conversions. | | | | |

## 2044h: Analog Input Parameters

| 2044.01h | Analog Input 1 Offset | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | [$-2^{(15)}$] - [$2^{(15)}$ –1] | DVU | Read / Write | Yes |
| **Description:**<br>This sub index contains a value corresponding to the Analog Input 1 Offset.<br>To convert the desired Offset Voltage to the appropriate value for this object:<br>Multiply Voltage (in decimal) by 819.2 and ignore any resulting fractional part. Now convert this decimal value to hexidecimal. | | | | |

| 2044.02h | Analog Input 1 Scale Factor | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | [$-2^{(31)}$] - [$2^{(31)}$ –1] | - | Read / Write | Yes |

**Description:**
This sub index contains a value corresponding to the scale factor for analog input 1. The values contained in this object are mode dependent and require a different algorithm to calculate for each mode. The easiest way to determine the desired value for a given scale factor is to use the DriveWare300 setup software to set a scale factor in the I/O Configuration block and then read the value up from this object.

Assigned to Current Loop Example: Desired scale factor = (X Amps / 1 Volt)
(X Amps * 10 * 2^18) / Drive Peak Current = Value in decimal; convert to hex.

Assigned to Velocity Loop Example: Desired Scale factor = (X cnts/sec / 1 Volt)
Convert X cnts/sec → Y cnts/100us by dividing by 10000.
Now multiply: Ycnts * 20 * 2^18 = Value in Decimal; convert to hex.

Assigned to Position Loop Example: Desired Scale Factor = (X cnts / 1 Volt)
Now Multiply: X cnts * 80 = Value in Decimal; convert to hex.

Assigned to Current Limit Example: Desired Scale Factor = (X % of drive peak / 1 Volt)
Cannot achieve a value higher than 20% / 1 Volt.
Now Multiply X * 2^18 / 5 = Value in Decimal; convert to hex.

Assigned to External Temperature: Desired Scale Factor = (X degrees C / 1 Volt)
Now multiply X *20 *2^18 = Value in Decimal; convert to hex

| 2044.03h | Analog Input 2 Offset | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] - [2^{(15)} - 1]$ | DVU | Read / Write | Yes |

**Description:**
This sub index contains a value corresponding to the Analog Input 2 Offset.
To convert the desired Offset Voltage to the appropriate value for this object:
Multiply Voltage (in decimal) by 819.2 and ignore any resulting fractional part. Now convert this decimal value to hexidecimal.

| 2044.04h | Analog Input 2 Scale Factor | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] - [2^{(31)} - 1]$ | - | Read / Write | Yes |

**Description:**
This sub index contains a value corresponding to the scale factor for analog input 2. The values contained in this object are mode dependent and require a different algorithm to calculate for each mode. The easiest way to determine the desired value for a given scale factor is to use the DriveWare300 setup software to set a scale factor in the I/O Configuration block and then read the value up from this object.

Assigned to Current Loop Example: Desired scale factor = (X Amps / 1 Volt)
(X Amps * 10 * 2^18) / Drive Peak Current = Value in decimal; convert to hex.

Assigned to Velocity Loop Example: Desired Scale factor = (X cnts/sec / 1 Volt)
Convert X cnts/sec → Y cnts/100us by dividing by 10000.
Now multiply: Ycnts * 20 * 2^18 = Value in Decimal; convert to hex.

Assigned to Position Loop Example: Desired Scale Factor = (X cnts / 1 Volt)
Now Multiply: X cnts * 80 = Value in Decimal; convert to hex.

Assigned to Current Limit Example: Desired Scale Factor = (X % of drive peak / 1 Volt)
Cannot achieve a value higher than 20% / 1 Volt.
Now Multiply X * 2^18 / 5 = Value in Decimal; convert to hex.

Assigned to External Temperature: Desired Scale Factor = (X degrees C / 1 Volt)
Now multiply X *20 *2^18 = Value in Decimal; convert to hex

| 2044.05h | Analog Input 3 Offset | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] - [2^{(15)} - 1]$ | DVU | Read / Write | Yes |

**Description:**
This sub index contains a value corresponding to the Analog Input 3 Offset.
To convert the desired Offset Voltage to the appropriate value for this object:
Multiply Voltage (in decimal) by 819.2 and ignore any resulting fractional part. Now convert this decimal value to hexidecimal.

| 2044.06h | Analog Input 3 Scale Factor | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] - [2^{(31)} - 1]$ | - | Read / Write | Yes |

**Description:**
This sub index contains a value corresponding to the scale factor for analog input 3. The values contained in this object are mode dependent and require a different algorithm to calculate for each mode. The easiest way to determine the desired value for a given scale factor is to use the DriveWare300 setup software to set a scale factor in the I/O Configuration block and then read the value up from this object.

Assigned to Current Loop Example: Desired scale factor = (X Amps / 1 Volt)
(X Amps * 10 * 2^18) / Drive Peak Current = Value in decimal; convert to hex.

Assigned to Velocity Loop Example: Desired Scale factor = (X cnts/sec / 1 Volt)
Convert X cnts/sec → Y cnts/100us by dividing by 10000.
Now multiply: Ycnts * 20 * 2^18 = Value in Decimal; convert to hex.

Assigned to Position Loop Example: Desired Scale Factor = (X cnts / 1 Volt)
Now Multiply: X cnts * 80 = Value in Decimal; convert to hex.

Assigned to Current Limit Example: Desired Scale Factor = (X % of drive peak / 1 Volt)
Cannot achieve a value higher than 20% / 1 Volt.
Now Multiply X * 2^18 / 5 = Value in Decimal; convert to hex.

Assigned to External Temperature: Desired Scale Factor = (X degrees C / 1 Volt)
Now multiply X *20 *2^18 = Value in Decimal; convert to hex

| 2044.07h | Analog Input 4 Offset | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] - [2^{(15)} - 1]$ | DVU | Read / Write | Yes |

**Description:**
This sub index contains a value corresponding to the Analog Input 4 Offset.
To convert the desired Offset Voltage to the appropriate value for this object:
Multiply Voltage (in decimal) by 819.2 and ignore any resulting fractional part. Now convert this decimal value to hexidecimal.

| 2044.08h | Analog Input 4 Scale Factor | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] - [2^{(31)} - 1]$ | - | Read / Write | Yes |

**Description:**
This sub index contains a value corresponding to the scale factor for analog input 4. The values contained in this object are mode dependent and require a different algorithm to calculate for each mode. The easiest way to determine the desired value for a given scale factor is to use the DriveWare300 setup software to set a scale factor in the I/O Configuration block and then read the value up from this object.

Assigned to Current Loop Example: Desired scale factor = (X Amps / 1 Volt)
(X Amps * 10 * 2^18) / Drive Peak Current = Value in decimal; convert to hex.

Assigned to Velocity Loop Example: Desired Scale factor = (X cnts/sec / 1 Volt)
Convert X cnts/sec → Y cnts/100us by dividing by 10000.
Now multiply: Ycnts * 20 * 2^18 = Value in Decimal; convert to hex.

Assigned to Position Loop Example: Desired Scale Factor = (X cnts / 1 Volt)
Now Multiply: X cnts * 80 = Value in Decimal; convert to hex.

Assigned to Current Limit Example: Desired Scale Factor = (X % of drive peak / 1 Volt)
Cannot achieve a value higher than 20% / 1 Volt.
Now Multiply X * 2^18 / 5 = Value in Decimal; convert to hex.

Assigned to External Temperature: Desired Scale Factor = (X degrees C / 1 Volt)
Now multiply X *20 *2^18 = Value in Decimal; convert to hex

## 2045h: Interface Inputs

| 2045.01h | Interface Input 1 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] - [2^{(31)} - 1]$ | Volts | Read / Write | No |
| **Description:** Defines the value used with interface input 1. | | | | |

| 2045.02h | Interface Input 2 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] - [2^{(31)} - 1]$ | Volts | Read / Write | No |
| **Description:** Defines the value used with interface input 2. | | | | |

| 2045.03h | Interface Input 3 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] - [2^{(31)} - 1]$ | Volts | Read / Write | No |
| **Description:** Defines the value used with interface input 3. | | | | |

| 2045.04h | Interface Input 4 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] - [2^{(31)} - 1]$ | Volts | Read / Write | No |

**Description:**

Defines the value used with interface input 4.

## 2046h: Auxiliary Input Parameters

| 2046.01h | Auxiliary Input 1 Input Counts | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | - | Read / Write | Yes |

**Description:**
Contains a value corresponding to the number of input counts in the input/output ratio used for Encoder following and Step and Direction modes.

| 2046.02h | Auxiliary Input 1 Output Counts | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | - | Read / Write | Yes |

**Description:**
Contains a value corresponding to the output in the input/output ratio used for Encoder following and Step and Direction modes. Encoder following mode can be used only when the position loop is closed. However, Step and Direction can be used to control position, velocity or current. Therefore, the scaling value used is mode dependent.

| 2046.03h | Auxiliary Input Polarity | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which direction the motor travels in while operating in encoder following mode.

| 2046.04h | Auxiliary Input Preload Enumerator | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**

Defines what a ‘Load Target’ command does while operating in Step and Direction or Encoder Following Mode.

| Description | Value |
|---|---|
| Set Target to 0 | 0 |
| Set Target to non-zero value | 1 |
| Set Target equal to Measured Position | 2 |

| 2046.05h | Auxiliary Input Preset Command Value | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | [$-2^{(31)}$] - [$2^{(31)}$ –1] | See Table | Read / Write | Yes |

**Description:**
Contains a value for the target when the Load Target command is set to load a nonzero value (as defined in 2046.04). Encoder following mode can be used only when the position loop is closed. However, Step and Direction can be used to control position, velocity or current. Therefore, the scaling value used is mode dependent.

## 2048h: PVT Parameters

| 2048.01h | Buffer Threshold Warning Level | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |
| **Description:**<br>A buffer threshold warning will occur when this number of PVT points is left in the buffer. | | | | |

| 2048.02h | PVT Input Method | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines if incremental or absolute position is to be used with PVT commands. Incremental position sets the PVT target position point equal to the previous PVT position point plus the specified value. Absolute position sets the PVT target position point equal to the specified value.

| Input Method | Value |
|---|---|
| Absolute position with sequence counter | 0 |
| Incremental position with sequence counter | 1 |

## 2054h: Drive Temperature Parameters

| 2054.01h | Drive Analog Temperature Disable Level | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | [$-2^{(15)}$] - [$2^{(15)}$ –1] | DTU | Read / Write | Yes |

**Description:**

This sub index contains a value corresponding to the temperature disable level for the drive's internal analog temperature sensor. (Not all drives have adjustable temperature sensors. Some may be equipped with fixed-value sensors). See Appendix A for DTU (Drive Temperature Units).

| 2054.02h | Drive Analog Temperature Enable Level | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | [$-2^{(15)}$] - [$2^{(15)}$ –1] | DTU | Read / Write | Yes |

**Description:**

This sub index contains a value corresponding to the temperature enable level for the drive after an over temperature event. See Appendix A for DTU (Drive Temperature Units).

| 2054.03h | External Analog Temperature Disable Level | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | [$-2^{(31)}$] - [$2^{(31)}$ –1] | DTU | Read / Write | Yes |

**Description:**
This sub index contains a value corresponding to the temperature disable level for an analog over temperature event. See Appendix A for DTU (Drive Temperature Units).

| 2054.04h | External Analog Temperature Enable Level | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] - [2^{(31)} - 1]$ | DTU | Read / Write | Yes |
| **Description:** This sub index contains a value corresponding to the temperature re-enable level after the analog over temperature event has been activated. See Appendix A for DTU (Drive Temperature Units). | | | | |

## 2058h: Digital Input Parameters

**Table 24: Object 2058 Mapping**

| Bit | Digital Input Mask |
|---|---|
| 0 | Digital Input 1 |
| 1 | Digital Input 2 |
| 2 | Digital Input 3 |
| 3 | Digital Input 4 |
| 4 | Digital Input 5 |
| 5 | Digital Input 6 |
| 6 | Digital Input 7 |
| 7 | Digital Input 8 |
| 8…15 | Reserved |

* Number of actual inputs depends on drive model

| 2058.01h | Digital Input Active Level Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | $0 - [2^{(16)} - 1]$ | N/A | Read / Write | Yes |
| **Description:** Determines which digital inputs are active high and which are active low. See (*Table 24*) above for mapping structure. | | | | |

| 2058.02h | Digital Input Inhibit Bridge Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | $0 - [2^{(16)} - 1]$ | N/A | Read / Write | Yes |
| **Description:** Defines which digital inputs, if any, are assigned to Inhibit the Bridge. See (*Table 24*) above for mapping structure. | | | | |

| 2058.03h | Digital Input Positive Limit Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | $0 - [2^{(16)} - 1]$ | N/A | Read / Write | Yes |

**Description:**
Defines which digital inputs, if any, are assigned to the positive limit. See (*Table 24*) above for mapping structure.

| 2058.02h | Digital Input Negative limit Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital inputs, if any, are assigned to negative limit. See (*Table 24*) above for mapping structure.

| 2058.05h | Digital Input Motor Over Temperature Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital inputs, if any, are assigned to activate Motor Over Temperature. See (*Table 24*) above for mapping structure.

| 2058.06h | Digital Input Phase Detection Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital inputs, if any, are assigned to activate Phase Detection. See (*Table 24*) above for mapping structure.

| 2058.07h | Digital Input Dynamic Brake Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital inputs, if any, are assigned to activate the Dynamic Brake. See (*Table 24*) above for mapping structure.

| 2058.08h | Digital Input Load Measured Position Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital inputs, if any, are assigned to activate the Load Measured Position event. See (*Table 24*) above for mapping structure.

| 2058.09h | Digital Input Load Target Position Command Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

| |
|---|
| **Description:**<br>Defines which digital inputs, if any, are assigned to activate the Load Target event. See (*Table 24*) above for mapping structure. |

| 2058.0Ah | Digital Input Start Homing Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |
| **Description:**<br>Defines which digital inputs, if any, are assigned to activate the Start Homing event. See (*Table 24*) above for mapping structure. | | | | |

| 2058.0Bh | Digital Input Home Switch Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |
| **Description:**<br>Defines which digital inputs, if any, are assigned to the Home Switch. See (*Table 24*) above for mapping structure. | | | | |

| 2058.0Ch | Digital Input Stop Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |
| **Description:**<br>Defines which digital inputs, if any, are assigned to the Stop event. See (*Table 24*) above for mapping structure. | | | | |

### 205Ah: Digital Output Parameters

All listed sub indexes used with object index 205Ah utilize the mapping table shown below.

**Table 25: Object 205A Mapping**

| Bit | Digital Output Mask |
|---|---|
| 0 | Digital Output 1 |
| 1 | Digital Output 2 |
| 2 | Digital Output 3 |
| 3 | Digital Output 4 |
| 4…15 | Reserved |

| 205A.01h | Digital Output Active Level Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs are active high and which are active low. See (*Table 25*) above for mapping structure.

| 205A.02h | Digital Output Drive Reset Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Drive Reset event. See (*Table 25*) above for mapping structure.

| 205A.03h | Digital Output Drive Internal Error Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Drive Internal Error event. See (*Table 25*) above for mapping structure.

| 205A.04h | Digital Output Short Circuit Fault Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Short Circuit Fault event. See (*Table 25*) above for mapping structure.

| 205A.05h | Digital Output Over-Current Fault | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Over-Current event. See (*Table 25*) above for mapping structure.

| 205A.06h | Digital Output Hard Under Voltage Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Hardware Under Voltage event. See (*Table 25*) above for mapping structure.

| 205A.07h | Digital Output Hard Over Voltage Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Hardware Over Voltage event. See (*Table 25*) above for mapping structure.

| 205A.08h | Digital Output Drive Over Temperature Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Drive Over Temperature event. See (*Table 25*) above for mapping structure.

| 205A.09h | Digital Output Parameter Restore Error Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Parameter Restore Error event. See (*Table 25*) above for mapping structure.

| 205A.0Ah | Digital Output Parameter Store Error Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Parameter Store Error event. See (*Table 25*) above for mapping structure.

| 205A.0Bh | Digital Output Invalid Hall State Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Invalid Hall State event. See (*Table 25*) above for mapping structure.

| 205A.0Ch | Digital Output Phase Synchronization Error Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Phase Synchronization Error event. See (*Table 25*) above for mapping structure.

| 205A.0Dh | Digital Output Motor Over Temperature Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Motor Over Temperature event. See (*Table 25*) above for mapping structure.

| 205A.0Eh | Digital Output Phase Detection Fault Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Phase Detection Fault event. See (*Table 25*) above for mapping structure.

| 205A.0Fh | Digital Output Feedback Sensor Error Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Feedback Sensor Error event. See (*Table 25*) above for mapping structure.

| 205A.10h | Digital Output Log Entry Missed Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Log Entry Missed event. See (*Table 25*) above for mapping structure.

| 205A.11h | Digital Output Commanded Inhibit Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Commanded Inhibit event. See (*Table 25*) above for mapping structure.

| 205A.12h | Digital Output User Inhibit Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the User Inhibit event. See (*Table 25*) above for mapping structure.

| 205A.13h | Digital Output Positive Limit Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Positive Limit event. See (*Table 25*) above for mapping structure.

| 205A.14h | Digital Output Negative Limit Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Negative Limit event. See (*Table 25*) above for mapping structure.

| 205A.15h | Digital Output Current Limiting (Foldback) Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Current Limiting event. See (*Table 25*) above for mapping structure.

| 205A.16h | Digital Output Continuous Current Limit Reached Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Continuous Current Limit Reached event. See (*Table 25*) above for mapping structure.

| 205A.17h | Digital Output Current Loop Saturated Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Current Loop Saturated event. See (*Table 25*) above for mapping structure.

| 205A.18h | Digital Output User Under Voltage Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the User Under Voltage event. See (*Table 25*) above for mapping structure.

| 205A.19h | Digital Output User Over Voltage Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the User Over Voltage event. See (*Table 25*) above for mapping structure.

| 205A.1Ah | Digital Output Non-Sinusoidal Commutation Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Non-Sinusoidal Commutation. See (*Table 25*) above for mapping structure.

| 205A.1Bh | Digital Output Phase Detection Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Phase Detection event. See (*Table 25*) above for mapping structure.

| 205A.1Ch | Digital Output Commanded Dynamic Brake Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Commanded Dynamic Brake event. See (*Table 25*) above for mapping structure.

| 205A.1Dh | Digital Output User Dynamic Brake Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the User Dynamic Brake event. See (*Table 25*) above for mapping structure.

| 205A.1Eh | Digital Output Shunt Regulator Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Shunt Regulator event. See (*Table 25*) above for mapping structure.

| 205A.1Fh | Digital Output Phase Detection Complete Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Phase Detection Complete event. See (*Table 25*) above for mapping structure.

| 205A.20h | Digital Output Command Profiler Active Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Command Profiler Active event. See (*Table 25*) above for mapping structure.

| 205A.21h | Digital Output Motor Over Speed Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Motor Over Speed event. See (*Table 25*) above for mapping structure.

| 205A.22h | Digital Output At Command Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the At Command event. See (*Table 25*) above for mapping structure.

| 205A.23h | Digital Output Zero Velocity Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Zero Velocity event. See (*Table 25*) above for mapping structure.

| 205A.24h | Digital Output Velocity Following Error Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Velocity Following Error event. See (*Table 25*) above for mapping structure.

| 205A.25h | Digital Output Positive Velocity Limit Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Positive Velocity Limit event. See (*Table 25*) above for mapping structure.

| 205A.26h | Digital Output Negative Velocity Limit Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Negative Velocity Limit event. See (*Table 25*) above for mapping structure.

| 205A.27h | Digital Output Max Measured Position Limit Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Max Measured Position event. See (*Table 25*) above for mapping structure.

| 205A.28h | Digital Min Measured Position Limit Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Min Measured Position event. See (*Table 25*) above for mapping structure.

| 205A.29h | Digital Output At Home Position Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the At Home Position event. See (*Table 25*) above for mapping structure.

| 205A.2Ah | Digital Output Position Following Error Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Position Following Error event. See (*Table 25*) above for mapping structure.

| 205A.2Bh | Digital Output Max Target position Limit Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Max Target Position Limit event. See (*Table 25*) above for mapping structure.

| 205A.2Ch | Digital Output Min Target Position Limit Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Min Target Position Limit event. See (*Table 25*) above for mapping structure.

| 205A.2Dh | Digital Output Load Measured Position Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Load Measured Position event. See (*Table 25*) above for mapping structure.

| 205A.2Eh | Digital Output Load Target Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Load Target event. See (*Table 25*) above for mapping structure.

| 205A.2Fh | Digital Output Homing Active Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Homing Active event. See (*Table 25*) above for mapping structure.

| 205A.30h | Digital Output Apply Brake Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Apply Brake event. See (*Table 25*) above for mapping structure.

| 205A.31h | Digital Output PVT Buffer Full Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Writ | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the PVT Buffer Full event. See (*Table 25*) above for mapping structure.

| 205A.32h | Digital Output PVT Buffer Empty Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the PVT Buffer Empty event. See (*Table 25*) above for mapping structure.

| 205A.33h | Digital Output PVT Buffer Threshold Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the PVT Buffer Threshold event. See (*Table 25*) above for mapping structure.

| 205A.34h | Digital Output PVT Buffer Failure Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the PVT Buffer Failure event. See (*Table 25*) above for mapping structure.

| 205A.35h | Digital Output PVT Buffer Empty Stop Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the PVT Buffer Empty Stop event. See (*Table 25*) above for mapping structure.

| 205A.36h | Digital Output PVT Sequence Number Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the PVT Sequence Number event. See (*Table 25*) above for mapping structure.

| 205A.37h | Digital Output Communication Error Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Communication Error event. See (*Table 25*) above for mapping structure.

| 205A.38h | Digital Output Homing Complete Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Homing Complete event. See (*Table 25*) above for mapping structure.

| 205A.39h | Digital Output Commanded Stop Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Commanded Stop event. See (*Table 25*) above for mapping structure.

| 205A.3A h | Digital Output User Stop Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the User Stop event. See (*Table 25*) above for mapping structure.

| 205A.3B h | Digital output bridge enabled mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the User Stop event. See (*Table 25*) above for mapping structure.

| 205A.3C h | Digital output dynamic brake active mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Dynamic Brake Active event. See (*Table 25*) above for mapping structure.

| 205A.3D h | Digital output stop active mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Stop Active event. See (*Table 25*) above for mapping structure.

| 205A.3E h | Digital output positive stop active mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Positive Stop Active event. See (*Table 25*) above for mapping structure.

| 205A.3F h | Digital output negative stop active mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Negative Stop Active event. See (*Table 25*) above for mapping structure.

| 205A.40 h | Digital output positive inhibit active mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Positive Inhibit Active event. See (*Table 25*) above for mapping structure.

| 205A.41 h | Digital output negative inhibit active mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to the Negative Inhibit Active event. See (*Table 25*) above for mapping structure.

| 205A.42 h | Digital Output Virtual Output 1 Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to Virtual Output 1. See (*Table 25*) above for mapping structure.

| 205A.43 h | Digital Output Virtual Output 2 Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to Virtual Output 2. See (*Table 25*) above for mapping structure.

| 205A.44 h | Digital Output Virtual Output 3 Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to Virtual Output 3. See (*Table 25*) above for mapping structure.

| 205A.45 h | Digital Output Virtual Output 4 Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to Virtual Output 4. See (*Table 25*) above for mapping structure.

| 205A.46 h | Digital Output Virtual Output 5 Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to Virtual Output 5. See (*Table 25*) above for mapping structure.

| 205A.47 h | Digital Output Virtual Output 6 Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to Virtual Output 6. See (*Table 25*) above for mapping structure.

| 205A.48 h | Digital Output Virtual Output 7 Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to Virtual Output 7. See (*Table 25*) above for mapping structure.

| 205A.49 h | Digital Output Virtual Output 8 Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to Virtual Output 8. See (*Table 25*) above for mapping structure.

| 205A.4B h | Digital Output Virtual Output 9 Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to Virtual Output 9. See (*Table 25*) above for mapping structure.

| 205A.4C h | Digital Output Virtual Output 10 Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to Virtual Output 10. See (*Table 25*) above for mapping structure.

| 205A.4D h | Digital Output Virtual Output 11 Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to Virtual Output 11. See (*Table 25*) above for mapping structure.

| 205A.4E h | Digital Output Virtual Output 12 Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to Virtual Output 12. See (*Table 25*) above for mapping structure.

| 205A.4F h | Digital Output Virtual Output 13 Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

**Description:**
Defines which digital outputs, if any, are assigned to Virtual Output 13. See (*Table 25*) above for mapping structure.

| 205A.50 h | Digital Output Virtual Output 14 Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |

| Description: |
|---|
| Defines which digital outputs, if any, are assigned to Virtual Output 14. See (*Table 25*) above for mapping structure. |

| 205A.51 h | Digital Output Virtual Output 15 Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |
| **Description:** Defines which digital outputs, if any, are assigned to Virtual Output 15. See (*Table 25*) above for mapping structure. | | | | |

| 205A.52 h | Digital Output Virtual Output 16 Mask | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | N/A | Read / Write | Yes |
| **Description:** Defines which digital outputs, if any, are assigned to Virtual Output 16. See (*Table 25*) above for mapping structure. | | | | |

## 205Ch: Analog Output Parameters

| 205C.01h | Analog Output 1 Signal Select A | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | - | Read / Write | Yes |
| **Description:** Together with Signal Select B determines which internal drive parameter is assigned to analog output 1. | | | | |

| 205C.02h | Analog Output 1 Signal Select B | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | - | Read / Write | Yes |
| **Description:** Together with Signal Select A determines which internal drive parameter is assigned to analog output 1. | | | | |

| 205C.03h | Analog Output 1 Offset | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | [$-2^{(15)}$] - [$2^{(15)}$ –1] | - | Read / Write | Yes |
| **Description:** Analog output 1 offset. | | | | |

| 205C.04h | Analog Output 1 Gain | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | [$-2^{(31)}$] - [$2^{(31)}$ –1] | - | Read / Write | Yes |
| **Description:** Analog output 1 gain. | | | | |

| 205C.05h | Analog Output 1 Operator | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | - | Read / Write | Yes |
| **Description:**<br>Analog output 1 operator. | | | | |

| 205C.06h | Analog Output 2 Signal Select A | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | - | Read / Write | Yes |
| **Description:**<br>Together with Signal Select B determines which internal drive parameter is assigned to analog output 2. | | | | |

| 205C.07h | Analog Output 2 Signal Select B | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | - | Read / Write | Yes |
| **Description:**<br>Together with Signal Select B determines which internal drive parameter is assigned to analog output 2. | | | | |

| 205C.08h | Analog Output 2 Offset | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | [$-2^{(15)}$] - [$2^{(15)}$ –1] | - | Read / Write | Yes |
| **Description:**<br>Analog output 2 offset. | | | | |

| 205C.09h | Analog Output 2 Gain | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | [$-2^{(31)}$] - [$2^{(31)}$ –1] | - | Read / Write | Yes |
| **Description:**<br>Analog output 2 gain. | | | | |

| 205C.0Ah | Analog Output 2 Operator | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | | Read / Write | Yes |
| **Description:**<br>Analog output 2 operator. | | | | |

## 2062h: Braking/Stop General Properties

| 2062.01h | Braking: Delay After Applying Brake | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br><br>Specifies the delay, in milliseconds, after applying the external brake before disabling the power bridge or dynamic braking. | | | | |

| 2062.02h | Braking: Delay Before Disengaging Brake | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0- [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br><br>Specifies the delay, in milliseconds, before releasing the external brake after enabling the power bridge or discontinuing dynamic braking. | | | | |

| 2062.03h | Quick Stop Deceleration Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | [$-2^{(31)}$] - [$2^{(31)}$ –1] | DAU | Read / Write | Yes |
| **Description:**<br><br>Specifies the maximum deceleration during a controlled stop event (quick stop). See Appendix A for DAU (Drive Acceleration Units). | | | | |

## 2064h: Fault Response Time Parameters

| 2064.01h | Fault Response Time Motor Over Temperature | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of Motor Over Temperature before its Fault Event Action (2065h) is executed. | | | | |

| 2064.02h | Fault Response Time Feedback Sensor Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of a Feedback Sensor Error before its Fault Event Action (2065h) is executed. | | | | |

| 2064.03h | Fault Response Time Log Entry Missed | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of a Log Entry Missed before its Fault Event Action (2065h) is executed. | | | | |

| 2064.04h | Fault Response Time User Inhibit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of a User Inhibit before the power bridge is disabled. | | | | |

| 2064.05h | Fault Response Time Positive Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of a Positive Limit input before its Fault Event Action (2065h) is executed. | | | | |

| 2064.06h | Fault Response Time Negative Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of a Negative Limit input before its Fault Event Action (2065h) is executed. | | | | |

| 2064.07h | Fault Response Time Current Limiting | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of Current Limiting before its Fault Event Action (2065h) is executed. | | | | |

| 2064.08h | Fault Response Time Continuous Current | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of reaching the Continuous Current setting before its Fault Event Action (2065h) is executed. | | | | |

| 2064.09h | Fault Response Time Current Loop Saturated | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of Current Loop Saturated before its Fault Event Action (2065h) is executed. | | | | |

| 2064.0Ah | Fault Response Time User Under Voltage | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of User Under Voltage before its Fault Event Action (2065h) is executed. | | | | |

| 2064.0Bh | Fault Response Time User Over Voltage | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time delay after the occurrence of a user-specified Over Voltage level before its Fault Event Action (2065h) is executed.

| 2064.0Ch | Fault Response Time Non-Sinusoidal Commutation | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time delay after the occurrence of Non-Sinusoidal Commutation before it is considered a Non-Sinusoidal Commutation.

| 2064.0Dh | Fault Response Time User Dynamic Brake | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time delay after the occurrence of a User Dynamic Brake input before dynamic braking is applied.

| 2064.0Eh | Fault Response Time Shunt Regulator | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time delay after the occurrence of Shunt Regulator activity before its Fault Event Action (2065h) is executed.

| 2064.0Fh | Fault Response Time Command Profiler Active | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time delay after the occurrence of Command Profiler Active before its Fault Event Action (2065h) is executed.

| 2064.10h | Fault Response Time At Command | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time delay after the occurrence of At Command before its Fault Event Action (2065h) is executed.

| 2064.11h | Fault Response Time Zero Velocity | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time delay after the occurrence of Zero Velocity before its Fault Event Action (2065h) is executed.

| 2064.12h | Fault Response Time Velocity Following Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of Velocity Following Error before its Fault Event Action (2065h) is executed. | | | | |

| 2064.13h | Fault Response Time Positive Velocity Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of Positive Velocity Limit before its Fault Event Action (2065h) is executed. | | | | |

| 2064.14h | Fault Response Time Negative Velocity Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of Negative Velocity Limit before its Fault Event Action (2065h) is executed. | | | | |

| 2064.15h | Fault Response Time At Home Position | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of At Home Position before its Fault Event Action (2065h) is executed. | | | | |

| 2064.16h | Fault Response Time Position Following Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of Position Following Error before its Fault Event Action (2065h) is executed. | | | | |

| 2064.17h | Fault Response Time Max Target Position Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of Max Target Position Limit before its Fault Event Action (2065h) is executed. | | | | |

| 2064.18h | Fault Response Time Min Target Position Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of Min Target Position Limit before its Fault Event Action (2065h) is executed. | | | | |

| 2064.19h | Fault Response Time Load Measured Position | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of Load Measured Position before defining the current position as a preset value (2039.01h). | | | | |

| 2064.1Ah | Fault Response Time Load Target | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of Load Target before the preset Target position is used. (2046h) | | | | |

| 2064.1Bh | Fault Response Time Homing Active | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of Homing Active before activating Homing. | | | | |

| 2064.1Ch | Fault Response Time PVT Buffer Full | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of PVT Buffer Full before its Fault Event Action (2065h) is executed. | | | | |

| 2064.1Dh | Fault Response Time PVT Buffer Empty | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after the occurrence of PVT Buffer Empty before its Fault Event Action (2065h) is executed. | | | | |

| 2064.1Eh | Fault Response Time PVT Buffer Threshold | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time delay after the occurrence of PVT Buffer Threshold before its Fault Event Action (2065h) is executed.

| 2064.1Fh | Fault Response Time PVT Buffer Failure | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time delay after the occurrence of PVT Buffer Failure before its Fault Event Action (2065h) is executed.

| 2064.20h | Fault Response Time PVT Buffer Empty Stop | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time delay after the occurrence of PVT Buffer Empty Stop before its Fault Event Action (2065h) is executed.

| 2064.21h | Fault Response Time PVT Sequence Number | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time delay after the occurrence of PVT Sequence Number before its Fault Event Action (2065h) is executed.

| 2064.22h | Fault Response Time Communication Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time delay after the occurrence of Communication Error before its Fault Event Action (2065h) is executed.

| 2064.23h | Fault Response Time User Stop | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time delay after the occurrence of a User Stop command before stopping the motor.

## 2065h: Fault Event Action Parameters

| 2065.01h | Fault Event Action Parameter Restore Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Parameter Restore Error. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.02h | Fault Event Action Parameter Store Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Parameter Store Error. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.03h | Fault Event Action Invalid Hall State | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after an Invalid Hall State. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.04h | Fault Event Action Phase Synch Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Phase Synch Error. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.05h | Fault Event Action Motor Over Temperature | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Motor Over Temperature. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.06h | Fault Event Action Feedback Sensor Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Feedback Sensor Error. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.07h | Fault Event Action Log Entry Missed | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Log Entry Missed. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.08h | Fault Event Action Current Limiting | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Current Limiting. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.09h | Fault Event Action Continuous Current | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Continuous Current. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.0Ah | Fault Event Action Current Loop Saturated | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after Current Loop Saturated. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.0Bh | Fault Event Action User Under Voltage | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a User Under Voltage. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.0Ch | Fault Event Action User Over Voltage | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a User Over Voltage. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.0Dh | Fault Event Action Shunt Regulator | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after Shunt Regulator active. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.0Eh | Fault Event Action Command Profiler Active | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after Command Profiler Active. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.0Fh | Fault Event Action Motor Over Speed | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Motor Over Speed. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.10h | Fault Event Action At Command | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after an At Command state. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.11h | Fault Event Action Zero Velocity | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Zero Velocity state. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.12h | Fault Event Action Velocity Following Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Velocity Following Error. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.13h | Fault Event Action Positive Velocity Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Positive Velocity Limit. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.14h | Fault Event Action Negative Velocity Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Negative Velocity Limit. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.15h | Fault Event Action Max Measured Position Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Max Measured Position Limit. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.16h | Fault Event Action Min Measured Position Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Min Measured Position Limit. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.17h | Fault Event Action At Home Position | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after an At Home Position state. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.18h | Fault Event Action Position Following Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Position Following Error. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.19h | Fault Event Action Max Target Position Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Max Target Position Limit. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.1Ah | Fault Event Action Min Target Position Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Min Target Position Limit. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.1Bh | Fault Event Action PVT Buffer Full | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a PVT Buffer Full status. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.1Ch | Fault Event Action PVT Buffer Empty | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a PVT Buffer Empty status. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.1Dh | Fault Event Action PVT Buffer Threshold | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after reaching PVT Buffer Threshold. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.1Eh | Fault Event Action PVT Buffer Failure | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a PVT Buffer Failure. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.1Fh | Fault Event Action PVT Buffer Empty Stop | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a PVT Buffer Empty Stop. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.20h | Fault Event Action PVT Sequence Number | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a PVT Sequence Number. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.21h | Fault Event Action Communication Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Communication Error. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.22h | Fault Event Action Positive Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Positive Limit. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.23h | Fault Event Action Negative Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Negative Limit. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.24h | Fault Event Action Drive Reset | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Drive Reset. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.25h | Fault Event Action Drive Internal Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Drive Internal Error. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.26h | Fault Event Action Short Circuit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Short Circuit. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.27h | Fault Event Action Current Overshoot | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Current Overshoot. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.28h | Fault Event Action Drive Under Voltage | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Drive Under Voltage. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.29h | Fault Event Action Drive Over Voltage | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Drive Over Voltage. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.2Ah | Fault Event Action Drive Over Temperature | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Drive Over Temperature. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.2Bh | Fault Event Action Commanded Inhibit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Commanded Inhibit. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.2Ch | Fault Event Action User Inhibit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a User Inhibit. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.2Dh | Fault Event Action Commanded Dynamic Brake | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Commanded Dynamic Brake. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.2Eh | Fault Event Action User Dynamic Brake | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a User Dynamic Brake. Refer to the table below (Table 26) for the valid event actions and their respective values.

| 2065.2Fh | Fault Event Action Phase Detection Fault | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read / Write | Yes |

**Description:**
The action of the drive immediately after a Phase Detection Fault. Refer to the table below (Table 26) for the valid event actions and their respective values.

**Table 26: Object 2065 Event Action Options**

| Sub Index | Object 2065h Event | Valid Event Action Values (refer to Table 27 for value definitions) | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 01h | Parameter Restore Error | - | 1 | - | - | 4 | - | - | - | 8 | 9 | 10 | - | - | - |
| 02h | Parameter Store Error | - | 1 | - | - | 4 | - | - | - | 8 | 9 | 10 | - | - | - |
| 03h | Invalid Hall State | - | 1 | - | - | 4 | - | - | - | 8 | 9 | 10 | - | - | - |
| 04h | Phase Synch Error | 0 | 1 | - | - | 4 | - | - | - | 8 | 9 | 10 | - | - | - |
| 05h | Motor Over Temperature | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 06h | Feedback Sensor Error | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 07h | Log Entry Missed | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 08h | Current Limiting | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 09h | Continuous Current | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 0Ah | Current Loop Saturated | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 0Bh | User Under Voltage | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 0Ch | User Over Voltage | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 0Dh | Shunt Regulator | 0 | 1 | - | - | 4 | - | - | - | 8 | 9 | 10 | - | - | - |
| 0Eh | Command Profiler Active | 0 | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 0Fh | Motor Over Speed | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 10h | At Command | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 11h | Zero Velocity | 0 | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 12h | Velocity Following Error | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 13h | Positive Velocity Limit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 14h | Negative Velocity Limit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 15h | Max Measured Position Limit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 16h | Min Measured Position Limit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 17h | At Home Position | 0 | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 18h | Position Following Error | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 19h | Max Target Position Limit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 1Ah | Min Target Position Limit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 1Bh | PVT Buffer Full | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 1Ch | PVT Buffer Empty | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 1Dh | PVT Buffer Threshold | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 1Eh | PVT Buffer Failure | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 1Fh | PVT Buffer Empty Stop | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 20h | PVT Sequence Number | 0 | 1 | 2 | 3 | 4 | - | - | - | 8 | 9 | 10 | - | - | - |
| 21h | Communication Error | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - | - | - |
| 22h | Positive Limit | - | - | 2 | - | - | 5 | - | - | - | - | - | - | - | - |
| 23h | Negative Limit | - | - | - | 3 | - | - | 6 | - | - | - | - | - | - | - |
| 24h | Drive Reset | - | 1 | - | - | - | - | - | - | - | - | - | - | - | - |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 25h | Drive Internal Error | - | 1 | - | - | - | - | - | - | - | - | - | - | - | - |
| 26h | Short Circuit | - | 1 | - | - | - | - | - | - | - | - | 10 | - | - | - |
| 27h | Current Overshoot | - | 1 | - | - | - | - | - | - | - | - | 10 | - | - | - |
| 28h | Drive Under Voltage | - | 1 | - | - | - | - | - | - | - | - | 10 | - | - | - |
| 29h | Drive Over Voltage | - | 1 | - | - | - | - | - | - | - | - | 10 | - | - | - |
| 2Ah | Drive Over Temperature | - | 1 | - | - | - | - | - | - | - | - | 10 | - | - | - |
| 2Bh | Commanded Inhibit | - | 1 | - | - | - | - | - | - | 8 | - | 10 | - | - | - |
| 2Ch | User Inhibit | - | 1 | - | - | - | - | - | - | 8 | - | 10 | - | - | - |
| 2Dh | Commanded Dynamic Brake | - | - | - | - | 4 | - | - | - | - | 9 | - | - | - | - |
| 2Eh | User Dynamic Brake | - | - | - | - | 4 | - | - | - | - | 9 | - | - | - | - |
| 2Fh | Phase Detection Fault | - | - | - | - | - | - | - | - | - | - | - | - | 12 | 15 |

**Table 27: Event Action Values Definition**

| Event Action Values | Hex Values | Event Actions |
|---|---|---|
| 0 | 00h | No Action |
| 1 | 01h | Disable Power Bridge |
| 2 | 02h | Disable Positive Direction |
| 3 | 03h | Disable Negative Direction |
| 4 | 04h | Dynamic Brake |
| 5 | 05h | Positive Stop |
| 6 | 06h | Negative Stop |
| 7 | 07h | Stop |
| 8 | 08h | Apply Brake **then** Disable Bridge |
| 9 | 09h | Apply Brake **then** Dynamic Brake |
| 10 | 0Ah | Apply Brake **and** Disable Bridge |
| 11 | 0Bh | Apply Brake **and** Dynamic Brake |
| 12 | 0Ch | Phase detect fault **and** disable bridge |
| 15 | 0Fh | Phase detect fault |

## 2066h: Fault Recovery Time Parameters

| 2066.01h | Fault Recovery Time Motor Over Temperature | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:** The time delay after Motor Over Temperature is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.02h | Fault Recovery Time Feedback Sensor Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:** The time delay after Feedback Sensor Error is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.03h | Fault Recovery Time Log Entry Missed | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:** The time delay after Log Entry Missed status is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.04h | Fault Recovery Time Commanded Inhibit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:** The time delay after Commanded Inhibit is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.05h | Fault Recovery Time User Inhibit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:** The time delay after User Inhibit is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.06h | Fault Recovery Time Positive Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:** The time delay after Positive Limit is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.07h | Fault Recovery Time Negative Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:** The time delay after Negative Limit is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.08h | Fault Recovery Time Current Limiting | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |

| | | | | |
|---|---|---|---|---|
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after Current Limiting is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.09h | Fault Recovery Time Continuous Current Limiting | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after Continuous Current Limiting is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.0Ah | Fault Recovery Time Current Loop Saturated | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after Current Loop Saturated status is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.0Bh | Fault Recovery Time User Under Voltage | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after User Under Voltage is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.0Ch | Fault Recovery Time User Over Voltage | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after User Over Voltage is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.0Dh | Fault Recovery Time Non-Sinusoidal Commutation | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after Non-Sinusoidal Commutation is no longer true before commutation is considered sinusoidal. | | | | |

| 2066.0Eh | Fault Recovery Time Phase Detection | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after Phase Detection state is no longer true before it is considered not in Phase Detection. | | | | |

| 2066.0Fh | Fault Recovery Time Commanded Brake | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

| Description: |
|---|
| The time delay after Commanded Brake is no longer true before it is considered no longer activated. |

| 2066.10h | Fault Recovery Time User Dynamic Brake | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:** The time delay after User Dynamic Brake is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.11h | Fault Recovery Time Shunt Regulator | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:** The time delay after Shunt Regulator active is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.12h | Fault Recovery Time Command Profiler Active | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:** The time delay after Command Profiler Active is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.13h | Fault Recovery Time Motor Over Speed | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:** The time delay after Motor Over Speed is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.14h | Fault Recovery Time At Command | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:** The time delay after At Command is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.15h | Fault Recovery Time Zero Velocity | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:** The time delay after Zero Velocity is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.16h | Fault Recovery Time Velocity Following Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:** The time delay after Velocity Following Error is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.17h | Fault Recovery Time Positive Velocity Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after Positive Velocity Limit is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.18h | Fault Recovery Time Negative Velocity Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after Negative Velocity Limit is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.19h | Fault Recovery Time Max Measured Position Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after Max Measured Position Limit status is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.1A | Fault Recovery Time Min Measured Position Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after Min Measured Position Limit status is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.1Bh | Fault Recovery Time At Home Position | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after no longer At Home Position before its Fault Event Action (2065h) is removed. | | | | |

| 2066.1Ch | Fault Recovery Time Position Following Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after Position Following Error is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.1Dh | Fault Recovery Time Max Target Position Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after Max Target Position Limit is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.1Eh | Fault Recovery Time Min Target Position Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |

| | | | | |
|---|---|---|---|---|
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after Min Target Position Limit is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.1Fh | Fault Recovery Time Load Measured Position | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after Load Measured Position is no longer true before it is considered no longer loading measured position. | | | | |

| 2066.20h | Fault Recovery Time Load Target | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after Load Target is no longer true before it is considered no longer loading target. | | | | |

| 2066.21h | Fault Recovery Time Homing Active | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after Homing Active status is no longer true before Homing is considered no longer active. | | | | |

| 2066.22h | Fault Recovery Time PVT Buffer Full | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after PVT Buffer Full is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.23h | Fault Recovery Time PVT Buffer Empty | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after PVT Buffer Empty is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.24h | Fault Recovery Time PVT Buffer Threshold | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after PVT Buffer Threshold is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.25h | Fault Recovery Time PVT Buffer Failure | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after PVT Buffer Failure is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.26h | Fault Recovery Time PVT Buffer Empty Stop | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after PVT Buffer Empty Stop is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.27h | Fault Recovery Time PVT Sequence Number | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after PVT Sequence Number error is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.28h | Fault Recovery Time Communication Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after Communication Error is no longer true before its Fault Event Action (2065h) is removed. | | | | |

| 2066.29h | Fault Recovery Time Commanded Stop | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after Commanded Stop is no longer true before it is considered no longer commanded. | | | | |

| 2066.2Ah | Fault Recovery Time User Stop | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time delay after User Stop is no longer true before it is considered no longer active. | | | | |

## 2067h: Fault Time-Out Window Parameters

| 2067.01h | Fault Time-Out Window Motor Over Temperature | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a Motor Over Temperature as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute. | | | | |

| 2067.02h | Fault Time-Out Window Feedback Sensor Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a Feedback Sensor Error as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.03h | Fault Time-Out Window User Inhibit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a User Inhibit as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.04h | Fault Time-Out Window Positive Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a Positive Limit as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.05h | Fault Time-Out Window Negative Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a Negative Limit as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.06h | Fault Time-Out Window Current Limiting | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of Current Limiting as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.07h | Fault Time-Out Window Continuous Current | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of Continuous Current as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.08h | Fault Time-Out Window Current Loop Saturated | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a Current Loop Saturated as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.09h | Fault Time-Out Window User Under Voltage | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a User Under Voltage as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.0Ah | Fault Time-Out Window User Over Voltage | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a User Over Voltage as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.0Bh | Fault Time-Out Window Non-Sinusoidal Commutation | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a Non-Sinusoidal Commutation as a new occurrence. The Event Action will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.0Ch | Fault Time-Out Window Phase Detection | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of Phase Detection as a new occurrence. The Event Action will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.0Dh | Fault Time-Out Window User Dynamic Brake | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a User Dynamic Brake as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute. | | | | |

| 2067.0Eh | Fault Time-Out Window Shunt Regulator | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a Shunt Regulator as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute. | | | | |

| 2067.0Fh | Fault Time-Out Window Command Profiler Active | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a Command Profiler Active as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute. | | | | |

| 2067.11h | Fault Time-Out Window Motor Over Speed | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a Motor Over Speed as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute. | | | | |

| 2067.12h | Fault Time-Out Window At Command | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of At Command as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute. | | | | |

| 2067.13h | Fault Time-Out Window Zero Velocity | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of Zero Velocity as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.14h | Fault Time-Out Window Velocity Following Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a Velocity Following Error as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.15h | Fault Time-Out Window Positive Velocity Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a Positive Velocity Limit as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.16h | Fault Time-Out Window Negative Velocity Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a Negative Velocity Limit as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.17h | Fault Time-Out Window Max Measured Position Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of Max Measured Position Limit as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.18h | Fault Time-Out Window Min Measured Position Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of Min Measured Position Limit as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.01h | Fault Time-Out Window At Home Position | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of At Home Position as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.19h | Fault Time-Out Window Position Following Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a Position Following Error as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.1Ah | Fault Time-Out Window Max Target Position Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of Max Target Position Limit as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.1Bh | Fault Time-Out Window Min Target Position Limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of Min Target Position Limit as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.1Ch | Fault Time-Out Window Load Measured Position | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of Load Measured Position as a new occurrence. The Event Action will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.1Dh | Fault Time-Out Window Load Target | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of Load Target as a new occurrence. The Event Action will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.1Eh | Fault Time-Out Window Homing Active | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a Homing Active as a new occurrence. The Event Action will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries attribute, which is unlimited for Homing.

| 2067.1Fh | Fault Time-Out Window PVT Buffer Full | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a PVT Buffer Full as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.20h | Fault Time-Out Window PVT Buffer Empty | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a PVT Buffer Empty as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.21h | Fault Time-Out Window PVT Buffer Threshold | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a PVT Buffer Threshold as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.22h | Fault Time-Out Window PVT Buffer Failure | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |

**Description:**
The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a PVT Buffer Failure as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute.

| 2067.23h | Fault Time-Out Window PVT Buffer Empty Stop | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a PVT Buffer Empty Stop as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute. | | | | |

| 2067.24h | Fault Time-Out Window PVT Sequence Number | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a PVT Sequence Number as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute. | | | | |

| 2067.25h | Fault Time-Out Window Communication Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a Communication Error as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute. | | | | |

| 2067.26h | Fault Time-Out Window User Stop | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – [$2^{(16)}$ –1] | Milliseconds | Read / Write | Yes |
| **Description:**<br>The time, after the Recovery Time (2066h) and subsequent removal of the event action, during which the drive will NOT consider an occurrence of a User Stop as a new occurrence. The Event Action (2065h) will still be applied in case an event does occur within this window. However, that occurrence will not be counted as a new occurrence with regard to the Maximum Recoveries (2068h) attribute. | | | | |

## 2068h: Fault Maximum Recoveries Parameters

| 2068.01h | Maximum Short Circuit Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |
| **Description:**<br>Each occurrence of a Short Circuit performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” and “Recovery Time”, a recovery counter is incremented. This object sets the maximum recovery count allowed before the Short Circuit fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software. | | | | |

| 2068.02h | Maximum Hard Under Voltage Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of a Hard Under Voltage performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” and “Recovery Time”, a recovery counter is incremented. This object sets the maximum recovery count allowed before the Hard Under Voltage fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.03h | Maximum Hard Over Voltage Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of a Hard Over Voltage performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” and “Recovery Time”, a recovery counter is incremented. This object sets the maximum recovery count allowed before the Hard Over Voltage fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.04h | Maximum Drive Over Temperature Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of a Drive Over Temperature performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” and “Recovery Time”, a recovery counter is incremented. This object sets the maximum recovery count allowed before the Drive Over Temerature fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.05h | Maximum Invalid Hall State Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of an Invalid Hall State performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” and “Recovery Time”, a recovery counter is incremented. This object sets the maximum recovery count allowed before the Invalid Hall State fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.06h | Maximum Phase Synchronization Error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of a Phase Synchronization Error performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” and “Recovery Time”, a recovery counter is incremented. This object sets the maximum recovery count allowed before the Phase Synchronization Error fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.07h | Maximum Motor Over Temperature Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of a Motor Over Temperature performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Motor Over Temperature fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.08h | Maximum Phase Detection Failure Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of a Phase Detection Failure performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” and “Recovery Time”, a recovery counter is incremented. This object sets the maximum recovery count allowed before the Phase Detection Failure fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.09h | Maximum Feedback Sensor Error Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of a Feedback Sensor Error performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Feedback Sensor Error fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.0Ah | Maximum Log Entry Missed Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of a Log Entry Missed performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Log Entry Missed fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.0Bh | Maximum User Inhibit Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of a User Inhibit performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the User Inhibit fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.0Ch | Maximum Positive Limit Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of a Positive Limit performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Positive Limit fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.0Dh | Maximum Negative Limit Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of a Negative Limit performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Negative Limit fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.0Eh | Maximum Current Limiting Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of Current Limiting performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Current Limiting fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.0Fh | Maximum Continuous Current Limiting Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of Continuous Current Limiting performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Continuous Current Limiting fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.10h | Maximum Current Loop Saturated Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of Current Loop Saturated performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Current Loop Saturated fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.11h | Maximum User Under Voltage Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of a User Under Voltage performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the User Under Voltage fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.12h | Maximum User Over Voltage Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of a User Over Voltage performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the User Over Voltage fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.13h | Maximum Non Sinusoidal Commutation Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of Non Sinusoidal Commutation performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Non Sinusoidal Commutation fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.14h | Maximum Phase Detection Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of Phase Detection performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Phase Detection fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.15h | Maximum User Dynamic Brake Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of a User Dynamic Brake performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the User Dynamic Brake fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.16h | Maximum Shunt Regulator Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of a Shunt Regulator performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Shunt Regulator fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.17h | Maximum Phase Detection Complete Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of a Phase Detection Complete performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” and “Recovery Time”, a recovery counter is incremented. This object sets the maximum recovery count allowed before the Phase Detection Complete fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.18h | Maximum Command Profiler Active Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of a Command Profiler Active performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Command Profiler Active fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.19h | Maximum Motor Over Speed Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of a Motor Over Speed performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Motor Over Speed fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.1Ah | Maximum At Command Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of At Command performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the At Command fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.1Bh | Maximum Zero Velocity Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of Zero Velocity performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Zero Velocity fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.1Ch | Maximum Velocity Following Error Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of Velocity Following Error performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Velocity Following Error fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.1Dh | Maximum Positive Velocity Limit Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of Positive Velocity Limit performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Positive Velocity Limit fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.1Eh | Maximum Negative Velocity Limit Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of Negative Velocity Limit performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Negative Velocity Limit fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.1Fh | Maximum Max Measured Position Limit Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of Max Measured Position performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Max Measured Position fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.20h | Maximum Min Measured Position Limit Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of Min Measured Position performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Min Measured Position fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.21h | Maximum At Home Position Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of At Home Position performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the At Home Position fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.22h | Maximum Position Following Errors Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of Position Following Errors performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Position Following Errors fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.23h | Maximum Max Target Position Limit Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of Max Target Position performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Max Target Position fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.24h | Maximum Min Target Position Limit Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of Min Target Position performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Min Target Position fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.25h | Maximum Load Measured Position Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of Load Measured Position performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Load Measured Position fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.26h | Maximum Load Target Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of Load Target Position performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Load Target Position fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.27h | Maximum PVT Buffer Full Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of PVT Buffer Full performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the PVT Buffer Full fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.28h | Maximum PVT Buffer Empty Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of PVT Buffer Empty performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the PVT Buffer Empty fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.29h | Maximum PVT Buffer Threshold Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of PVT Buffer Threshold performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the PVT Buffer Threshold fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.2Ah | Maximum PVT Buffer Failure Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of PVT Buffer Failure performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the PVT Buffer Failure fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.2Bh | Maximum PVT Buffer Empty Stop Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of PVT Buffer Empty Stop performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the PVT Buffer Empty Stop fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.2Ch | Maximum PVT Sequence Number Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of PVT Buffer Sequence Number performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the PVT Buffer Sequence Number fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.2Dh | Maximum Communication Error Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

**Description:**
Each occurrence of Communication Error performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” (2067h) and “Recovery Time” (2066h), a recovery counter is incremented. This object sets the maximum recovery count allowed before the Communication Error fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software.

| 2068.2Eh | Maximum Homing Complete Recoveries | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 65535 | Number of Recoveries | Read / Write | Yes |

| Description: |
|---|
| Each occurrence of Homing Complete performs the event action assigned to this fault. Each time the fault is removed for longer than the addition of the values in the “Time Out Window” and “Recovery Time”, a recovery counter is incremented. This object sets the maximum recovery count allowed before the Homing Complete fault latches and must be actively reset in order to enable the bridge. Re-setting the recovery counter requires a connection to the AMC drive configuration software appropriate for this drive. For more information on event handling, see the Help file associated with the AMC drive configuration software. |

### 6065h: Position Following Error Window

| 6065h | Position Following Error Window | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | 0 - [$2^{(32)}$ –1] | DPU | Read / Write | Yes |
| **Description:** The maximum allowed position error (difference between demand position and measured position), prior to setting the “Position Following Error” event (active in position mode only). See Appendix A for DPU (Drive Position Units). | | | | |

### 6098h: Homing Method

| 6098h | Homing Method | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer8 | 1 – 35 | N/A | Read / Write | Yes |
| **Description:** There are 35 homing methods supported by AMC CANopen servo drives. See section “Homing” for details on each homing method. | | | | |

### 6099h: Homing Speeds

| 6099.01h | Speed During Search For Switch | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | 0 – ($2^{32}$-1) | DSU | Read / Write | Yes |
| **Description:** The value assigned to this object sets the speed during the first stage of Homing algorithms. See Appendix A for DSU (Drive Velocity Units). | | | | |

| 6099.02h | Speed During Search For Zero | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | 0 – ($2^{32}$-1) | DSU | Read / Write | Yes |
| **Description:** The value assigned to this object sets the speed during the search for zero. This is usually after the search for switch has completed and is set much slower for accuracy. See Appendix A for DSU (Drive Velocity Units). | | | | |

### 609Ah: Homing Acceleration

| 609Ah | Homing Acceleration | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | 0 – ($2^{32}$-1) | DAU | Read / Write | Yes |

**Description:**
This object sets the accelerations and decelerations used by the drive's homing routine. See Appendix A for DAU (Drive Acceleration Units).

The following objects are typically used during operation. They are either used to perform specific tasks or to obtain information from the drive.

## 2002h: Drive Status

| 2002.01h | Drive Bridge Status | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | N/A | N/A | Read Only | No |
| **Description:** The function of each bit in this object is given in Table 28 below. | | | | |

| 2002.02h | Drive Protection Status | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | N/A | N/A | Read Only | No |
| **Description:** The function of each bit in this object is given in Table 28 below. | | | | |

| 2002.03h | System Protection Status | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | N/A | N/A | Read Only | No |
| **Description:** The function of each bit in this object is given in Table 28 below. | | | | |

| 2002.04h | Drive/System Status 1 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | N/A | N/A | Read Only | No |
| **Description:** The function of each bit in this object is given in Table 28 below. | | | | |

| 2002.05h | Drive/System Status 2 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | N/A | N/A | Read Only | No |
| **Description:** The function of each bit in this object is given in Table 28 below. | | | | |

| 2002.06h | Drive/System Status 3 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | N/A | N/A | Read Only | No |
| **Description:** The function of each bit in this object is given in Table 28 below. | | | | |

**Table 28**

| Bit | Drive Bridge Status | Drive Protection Status | System Protection Status | Drive System Status 1 | Drive System Status 2 | Drive System Status 3 |
|---|---|---|---|---|---|---|
| 0 | Bridge Enabled | Drive Reset | Parameter Restore Error | Log Entry Missed | Zero Velocity | PVT Buffer Full |

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | Brake Enabled | Drive Internal Error | Parameter Store Error | Commanded Inhibit | At Command | PVT Buffer Empty |
| 2 | Shunt Enabled | Short Circuit | Invalid Hall State | User Inhibit | Velocity Following Error | PVT Buffer Threshold |
| 3 | Stop Enabled | Current Overshoot | Phase Sync. Error | Positive Inhibit | Positive Target Velocity Limit | PVT Buffer Failure |
| 4 | Reserved | Under Voltage | Motor Over Temperature | Negative Inhibit | Negative Target Velocity Limit | PVT Buffer Empty Stop |
| 5 | Reserved | Over Voltage | Phase Detection Fault | Current Limiting | Command Profiler Active | PVT Buffer Sequence Error |
| 6 | Reserved | Drive Over Temperature | Feedback Sensor Error | Continuous Current | In Home Position | Commanded Stop |
| 7 | Reserved | Reserved | Motor Over Speed | Current Loop Saturated | Position Following Error | User Stop |
| 8 | Reserved | Reserved | Max Measured Position | User Under Voltage | Max Target Position Limit | Reserved |
| 9 | Reserved | Reserved | Min Measured Position | User Over Voltage | Min Target Position Limit | Reserved |
| 10 | Reserved | Reserved | Communication Error (Node Guarding) | Non-sinusoidal Commutation | Load Measured Position | Reserved |
| 11 | Reserved | Reserved | Reserved | Phase Detection | Load Target | Reserved |
| 12 | Reserved | Reserved | Reserved | Commanded Dynamic Brake | Homing Active | Reserved |
| 13 | Reserved | Reserved | Reserved | User Dynamic Brake | Apply Brake | Reserved |
| 14 | Reserved | Reserved | Reserved | Shunt Regulator | Homing Complete | Reserved |
| 15 | Reserved | Reserved | Reserved | Phase Detection Complete | Reserved | Reserved |

## 2003h: Drive Status History

| 2003.01h | Drive Bridge Status History | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | N/A | N/A | Read Only | No |
| **Description:**<br>If an event becomes active and then becomes inactive, Drive Status History will mark the event with a history bit. If a bit is 1, that event has occurred sometime in the past; 0 indicates the event has never occurred sice power-up. Any history bit can be cleared by writing a 1 to that bit.<br>The function of each bit in this object is given in Table 28 of object 2002h. | | | | |

| 2003.02h | Drive Protection Status History | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | N/A | N/A | Read Only | No |
| **Description:**<br>If an event becomes active and then becomes inactive, Drive Status History will mark the event with a history bit. If a bit is 1, that event has occurred sometime in the past; 0 indicates the event has never occurred sice power-up. Any history bit can be cleared by writing a 1 to that bit.<br>The function of each bit in this object is given in Table 28 of object 2002h. | | | | |

| 2003.03h | System Protection Status History | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | N/A | N/A | Read Only | No |

**Description:**
If an event becomes active and then becomes inactive, Drive Status History will mark the event with a history bit. If a bit is 1, that event has occurred sometime in the past; 0 indicates the event has never occurred sice power-up. Any history bit can be cleared by writing a 1 to that bit.
The function of each bit in this object is given in Table 28 of object 2002h.

| 2003.04h | Drive/System Status 1 History | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | N/A | N/A | Read Only | No |

**Description:**
If an event becomes active and then becomes inactive, Drive Status History will mark the event with a history bit. If a bit is 1, that event has occurred sometime in the past; 0 indicates the event has never occurred sice power-up. Any history bit can be cleared by writing a 1 to that bit.
The function of each bit in this object is given in Table 28 of object 2002h.

| 2003.05h | Drive/System Status 2 History | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | N/A | N/A | Read Only | No |

**Description:**
If an event becomes active and then becomes inactive, Drive Status History will mark the event with a history bit. If a bit is 1, that event has occurred sometime in the past; 0 indicates the event has never occurred sice power-up. Any history bit can be cleared by writing a 1 to that bit.
The function of each bit in this object is given in Table 28 of object 2002h.

| 2003.06h | Drive/System Status 3 History | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | N/A | N/A | Read Only | No |

**Description:**
If an event becomes active and then becomes inactive, Drive Status History will mark the event with a history bit. If a bit is 1, that event has occurred sometime in the past; 0 indicates the event has never occurred sice power-up. Any history bit can be cleared by writing a 1 to that bit.
The function of each bit in this object is given in Table 28 of object 2002h.

## 200Eh: Feedback Sensor Values

| 200E.01h | Primary encoder counts | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] – [2^{(31)}-1]$ | DPU | Read Only | No |

**Description:**

This object contains the current number of encoder counts from the primary encoder. It is an absolute value in that it does not depend on the current load measured position or home values.

| 200E.02h | Latched encoder/resolver position | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | $0 - [2^{(32)}–1]$ | DPU | Read Only | No |

**Description:**
This object contains a value corresponding to the latched encoder / resolver position.

| 200E.03h | Commutation synchronization counts | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] – [2^{(31)}-1]$ | DPU | Read Only | No |
| **Description:**<br>This object contains a value corresponding to the commutation synchronization counts. | | | | |

| 200E.04h | Hall sensor values | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | $0 – [2^{(16)} –1]$ | N/A | Read Only | No |
| **Description:**<br>This object contains a value corresponding to the Hall sensor values. | | | | |

## 200Fh: Power Bridge Values

| 200F.01h | DC bus voltage | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $0 – [2^{(15)}-1]$ | DVU | Read Only | No |
| **Description:**<br>This object contains a value corresponding to the DC Bus Voltage. See Appendix A for DVU (Drive Voltage Units). | | | | |

| 200F.02h | Phase A output voltage | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | DVU | Read Only | No |
| **Description:**<br>This object contains a value corresponding to the Phase A Output Voltage. See Appendix A for DVU (Drive Voltage Units). | | | | |

| 200F.03h | Phase B output voltage | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | DVU | Read Only | No |
| **Description:**<br>This object contains a value corresponding to the Phase B Output Voltage. See Appendix A for DVU (Drive Voltage Units). | | | | |

| 200F.04h | Phase C output voltage | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | DVU | Read Only | No |
| **Description:**<br>This object contains a value corresponding to the Phase C Output Voltage. See Appendix A for DVU (Drive Voltage Units). | | | | |

| 200F.05h | Trap mode output voltage | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | DVU | Read Only | No |
| **Description:**<br>This object contains a value corresponding to the trap mode output voltage. See Appendix A for DVU (Drive Voltage Units). | | | | |

## 2010h: Current Values

| 2010.01h | Current target - torque | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] – [2^{(31)}-1]$ | DCU | Read Only | No |
| **Description:** This object contains the current target torque when the drive is in Current (torque) mode. See Appendix A for DCU (Drive Current Units). | | | | |

| 2010.02h | Current demand - torque | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | DCU | Read Only | No |
| **Description:** This object contains the current demand torque when the drive is in Current (torque) mode. See Appendix A for DCU (Drive Current Units). | | | | |

| 2010.03h | Current measured - torque | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | DCU | Read Only | No |
| **Description:** This object contains the current measured torque when the drive is in Current (torque) mode. See Appendix A for DCU (Drive Current Units). | | | | |

| 2010.04h | Current error - torque | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | DCU | Read Only | No |
| **Description:** This object contains the error between the current target torque and the current measured torque. (Current Target Torque – Current Measured Torque = Value in 2010.04h). When the Target current is reached, the current error is zero. See Appendix A for DCU (Drive Current Units). | | | | |

| 2010.05h | Current target - flux | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] – [2^{(31)}-1]$ | DCU | Read Only | No |
| **Description:** This object contains a value corresponding to the current target flux. See Appendix A for DCU (Drive Current Units). | | | | |

| 2010.06h | Current demand - flux | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | DCU | Read Only | No |
| **Description:** This object contains a value corresponding to the current demand flux. See Appendix A for DCU (Drive Current Units). | | | | |

| 2010F.07h | Current measured - flux | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | DCU | Read Only | No |

**Description:**
This object contains a value corresponding to the current measured – flux. See Appendix A for DCU (Drive Current Units).

| 2010.08h | Current error - flux | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | DCU | Read Only | No |

**Description:**
This object contains a value corresponding to the Current error – flux. See Appendix A for DCU (Drive Current Units).

| 2010.09h | Current target - flux reference | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] – [2^{(31)}-1]$ | DCU | Read Only | No |

**Description:**
This object contains a value corresponding to the Current target flux reference. See Appendix A for DCU (Drive Current Units).

| 2010.0Ah | Current demand - flux reference | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | N/A | Read Only | No |

**Description:**
This object contains a value corresponding to the current demand flux reference.

| 2010.0Bh | Current measured - flux reference | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | N/A | Read Only | No |

**Description:**
This object contains a value corresponding to the current measured flux reference.

| 2010.0Ch | Current error - flux reference | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | N/A | Read Only | No |

**Description:**
This object contains a value corresponding to the current error flux reference.

| 2010.0Dh | Current limit | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] – [2^{(31)}-1]$ | N/A | Read Only | No |

**Description:**
This object contains a value corresponding to the current limit.

| 2010.0Eh | Current measured - phase A | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | DCU | Read Only | No |

**Description:**
This object contains a value corresponding to the current measured in phase A. See Appendix A for DCU (Drive Current Units).

| 2010.0Fh | Current measured - phase B | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | DCU | Read Only | No |
| **Description:**<br>This object contains a value corresponding to the current measured in phase B. See Appendix A for DCU (Drive Current Units). | | | | |

| 2010.10h | Phase angle - rotor | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 359 | DGU | Read Only | No |
| **Description:**<br>This object contains a value corresponding to the Phase Angle – Rotor. See Appendix A for DGU (Drive Angle Units). | | | | |

| 2010.11h | Phase angle - stator | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 359 | DGU | Read Only | No |
| **Description:**<br>This object contains a value corresponding to the Phase Angle – Stator. See Appendix A for DGU (Drive Angle Units). | | | | |

## 2011h: Velocity Values

| 2011.01h | Velocity measured pre-filter | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] – [2^{(31)}-1]$ | DSU | Read Only | No |
| **Description:**<br>This object contains the measured velocity before the feedback cutoff filter. See Appendix A for DSU (Drive Velocity Units). | | | | |

| 2011.02h | Velocity measured post-filter | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] – [2^{(31)}-1]$ | DSU | Read Only | No |
| **Description:**<br>This object contains the measured velocity after the feedback cutoff filter. See Appendix A for DSU (Drive Velocity Units). | | | | |

| 2011.03h | Velocity target | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] – [2^{(31)}-1]$ | DSU | Read Only | No |
| **Description:**<br>This object contains the current velocity target when the drive is in velocity mode. See Appendix A for DSU (Drive Velocity Units). | | | | |

| 2011.04h | Velocity demand | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] – [2^{(31)}-1]$ | DSU | Read Only | No |

| Description: |
|---|
| This object contains the current velocity demand when the drive is in velocity mode. See Appendix A for DSU (Drive Velocity Units). |

| 2011.05h | Velocity loop error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] – [2^{(31)}-1]$ | DSU | Read Only | No |
| **Description:** This object contains the error between the target velocity and the measured velocity. (Target Velocity – Measured Velocity = Value in 2011.05h). When the current commanded velocity is reached, the velocity loop error will be zero. See Appendix A for DSU (Drive Velocity Units). | | | | |

## 2012h: Position Values

| 2012.01h | Position measured | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] – [2^{(31)}-1]$ | DPU | Read Only | No |
| **Description:** This object contains the current measured position in counts. When the value in this object matches the value in object 2012.02h, the drive has reached the commanded position. See Appendix A for DPU (Drive Position Units). | | | | |

| 2012.02h | Position target | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] – [2^{(31)}-1]$ | DPU | Read Only | No |
| **Description:** This object contains the current commanded position when the drive is used in the position mode. See Appendix A for DPU (Drive Position Units). | | | | |

| 2012.03h | Position demand | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] – [2^{(31)}-1]$ | DPU | Read Only | No |
| **Description:** This object contains the current position demand in counts. See Appendix A for DPU (Drive Position Units). | | | | |

| 2012.04h | Position loop error | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] – [2^{(31)}-1]$ | DPU | Read Only | No |
| **Description:** This object contains the error between the target position (in counts) and the measured position (in counts). (Target Position (counts) – Measured Position (counts) = Value in 2012.04h). When the current commanded position is reached, the position loop error will be zero. See Appendix A for DPU (Drive Position Units). | | | | |

### 2014h: Command Profiler Input

| 2014.01h | Input command | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] – [2^{(31)}-1]$ | N/A | Read Only | No |
| **Description:** This object contains a value corresponding to the input of the command profiler. | | | | |

### 201Ah: Analog Input Values

| 201A.01h | Analog input value 1 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | DVU | Read Only | No |
| **Description:** This object contains a value corresponding to the voltage present on analog input 1. See Appendix A for DVU (Drive Voltage Units). | | | | |

| 201A.02h | Analog input value 2 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | DVU | Read Only | No |
| **Description:** This object contains a value corresponding to the voltage present on analog input 2. See Appendix A for DVU (Drive Voltage Units). | | | | |

| 201A.03h | Analog input value 3 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | DVU | Read Only | No |
| **Description:** This object contains a value corresponding to the voltage present on analog input 3. See Appendix A for DVU (Drive Voltage Units). | | | | |

| 201A.04h | Analog input value 4 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | DVU | Read Only | No |
| **Description:** This object contains a value corresponding to the voltage present on analog input 4. See Appendix A for DVU (Drive Voltage Units). | | | | |

### 201Ch: Auxiliary Input Values

| 201C.01h | Auxiliary input 1 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] – [2^{(31)}-1]$ | DPU | Read Only | No |
| **Description:** This object contains a value corresponding to the number of encoder counts are received from an auxiliary encoder that is used for feedback. See Appendix A for DPU (Drive Position Units). | | | | |

## 201Dh: PVT Status Values

| 201D.01h | PVT status | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | See Table | N/A | Read Only | No |

**Description:**
This object contains a value corresponding to the current status of a PVT move. The table below shows how to interpret the value in this object:

| Bit | PVT Status | Description |
|---|---|---|
| 0 | Buffer Full | The PVT Buffer is Full |
| 1 | Buffer Empty | The PVT Buffer is Empty |
| 2 | Buffer Threshold | The PVT Buffer has reached its threshold |
| 3 | Buffer Failure | Problem Reading Point From PVT Buffer |
| 4 | Buffer Empty Stop | The PVT Buffer is Empty, Last PVT Point has been reached |
| 5 | PVT point wrong sequence | A PVT Point Sequence Error as occurred |
| 6…15 | Reserved | Reserved For Future Use |

| 201D.02h | PVT Points Remaining | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read Only | No |

**Description:**

This object contains a value corresponding to the number of PVT points remaining in the PVT buffer. This value gets decremented by 1 after each PVT point is executed. When it reaches zero, the PVT buffer is empty.

| 201D.03h | PVT Sequence Number | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 – 15 | N/A | Read Only | No |

**Description:**
This object contains a value corresponding to the current PVT point in the PVT buffer that is being executed.

## 2021h: Drive Temperature Values

| 2021.01h | Drive base-plate temperature | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | DTU | Read Only | No |

**Description:**
This object contains a value corresponding to the temperature of the drive's base plate. See Appendix A for DTU (Drive Temperature Units).

| 2021.02h | External thermal sense value | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $[-2^{(31)}] – [2^{(31)}-1]$ | N/A | Read Only | No |

**Description:**
This object contains a value corresponding to the external thermal sense value

### 2023h: Digital Input Values

| 2023.01h | Digital inputs (post active level) | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | See Table | N/A | Read Only | No |

**Description:**
This object contains a value corresponding to the state of the drive's digital inputs. The table below shows the relationship between the drives digital inputs and the bit weights of the value in this object.

| Bit | Digital Inputs* |
|---|---|
| 0 | Digital Input 1 |
| 1 | Digital Input 2 |
| 2 | Digital Input 3 |
| 3 | Digital Input 4 |
| 4 | Digital Input 5 |
| 5 | Digital Input 6 |
| 6 | Digital Input 7 |
| 7 | Digital Input 8 |
| 8…15 | Reserved |

*Number of actual inputs depends on drive model

### 2024h: Digital Output Values

| 2024.01h | Digital outputs (pre active level) | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | See Table | N/A | Read Only | No |

**Description:**
This object contains a value corresponding to the state of the drive's digital outputs. The table below shows the relationship between the drives digital outputs and the bit weights of the value in this object.

| Bit | Digital Outputs |
|---|---|
| 0 | Digital Output 1 |
| 1 | Digital Output 2 |
| 2 | Digital Output 3 |
| 3 | Digital Output 4 |
| 4…15 | Reserved |

### 2025h: Analog Output Values

| 2025.01h | Analog output value 1 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | DVU | Read Only | No |

**Description:**
This object contains a value corresponding to the value of analog output 1. The analog outputs have a range of 0 to 10 Volts. See Appendix A for DVU (Drive Voltage Units).

| 2025.02h | Analog output value 2 | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $[-2^{(15)}] – [2^{(15)}-1]$ | DVU | Read Only | No |

**Description:**

This object contains a value corresponding to the value of analog output 2. The analog outputs have a range of 0 to 10 Volts. See Appendix A for DVU (Drive Voltage Units).

### 6040h: Controlword

| 6040h | Controlword | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 - 65535 | N/A | Read / Write | No |

**Description:**
The Controlword object sets the control state machine in the drive. State Machine Overview explains each drive state and how to use the Controlword to move the drive to that state. Below is a table providing the basic ControlWord commands.

| Value (Hex) | Command | Description |
|---|---|---|
| 80 | Reset Fault | On any transition to “1” of bit 7 causes a Rest Fault |
| 04 | Disable Voltage | Drive in “Switch On Disabled” state |
| 06 | Shutdown | Drive in “Not Ready to Switch On” state |
| 07 | Switch On | Drive in “Switched On” state |
| 0F | Enable Operation | Drive in “Operation Enabled” state |
| 1F | Start homing | Starts homing (when in homing mode) |
| 0F | End Homing | Ends homing |

### 6041h: Statusword

| 6041h | Statusword | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned16 | 0 - 655535 | N/A | Read Only | No |

**Description:**
The Statusword is used to determine which state the drive is in. Drive States explains each drive’s state and the Statusword bit definitions. Below is a table of the hex values for each state.

| Value | State | Description |
|---|---|---|
| xxxx xxxx x0xx 0000 | Not Ready to Switch On | Drive is initializing, drive is disabled |
| xxxx xxxx x1xx 0000 | Switch On Disabled | Drive completed initialization, drive is disabled |
| xxxx xxxx x01x 0001 | Ready to Switch On | Bus power may be applied, drive is disabled |
| xxxx xxxx x01x 0011 | Switched On | Bus power is applied, drive is disabled |
| xxxx xxxx x01x 0111 | Operation Enabled | Drive is enabled |
| xxxx xxxx x0xx 1000 | Fault | Drive is in the fault state |
| xxxx xxxx x00x 0111 | Quick Stop Active | Quick stop received from host and now in this state |

### 6060h: Modes Of Operation

| 6060h | Modes Of Operation | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer8 | -128 - 127 | N/A | Read / Write | No |

**Description:**
A “Mode Of Operation” refers to how the drives internal control loops are configured. Modes of Operation explains the valid control loop configurations for an AMC CANopen servo drive.

| Value | Operation Mode |
|---|---|
| 1 | Profile position mode |
| 3 | Profile velocity mode |
| 4 | Profile torque mode (current mode) |
| 6 | Homing mode |
| 7 | Interpolated position mode (PVT) |

### 6061h: Modes Of Operation Display

| 6061h | Modes Of Operation Display | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer8 | -128 - 127 | N/A | Read Only | No |

**Description:**
A “Mode Of Operation” refers to how the drives internal control loops are configured. Modes of Operation explains the valid control loop configurations for an AMC CANopen servo drive.

| Value | Operation Mode |
|---|---|
| 1 | Profile position mode |
| 3 | Profile velocity mode |
| 4 | Profile current mode |
| 6 | Homing mode |
| 7 | Interpolated position mode (PVT) |

### 6064h: Actual Position

| 6064h | Actual Position | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $-2^{31}$ – ($2^{31}$-1) | DPU | Read Only | No |
| **Description:**<br>Position Actual Value contains the measured position of the primary feedback device. This is the actual value used to create position error in profile position mode. See Appendix A for DPU (Drive Position Units). | | | | |

### 6069h: Velocity Sensor Actual Value

| 6069h | Velocity Sensor Actual Value | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $-2^{31}$ – ($2^{31}$-1) | DSU | Read Only | No |
| **Description:**<br>The value read from this object is the velocity measured directly from the primary feedback device before filtering or conditioning is applied. To read the actual velocity value used by the velocity control loop, see object 606C. See Appendix A for DSU (Drive Speed Units). | | | | |

### 606Bh: Velocity Demand

| 606Bh | Velocity Demand | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $-2^{31}$ – ($2^{31}$-1) | DSU | Read Only | No |
| **Description:**<br>Velocity Demand is defined as the target velocity, after limits and profiling, which is applied to the signal. This is the signal used by the velocity loop to produce a velocity error signal. See Appendix A for DSU (Drive Speed Units). | | | | |

### 606Ch: Actual Velocity

| 606Ch | Actual Velocity | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $-2^{31}$ – ($2^{31}$-1) | DSU | Read Only | No |

| Description: |
|---|
| Actual Velocity is defined as the measured velocity, after conditioning, used to close the drives velocity loop. See Appendix A for DSU (Drive Speed Units). |

### 6071h: Target Current

| 6071h | Target Current | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $-2^{15} – (2^{15}-1)$ | DCU | Read / Write | No |
| **Description:** Sets the Target Current while in Profile Current Mode (set by object 6060h). See Appendix A for DCU (Drive Current Units). | | | | |

### 6077h: Actual Current

| 6077h | Actual Current | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $-2^{15} – (2^{15}-1)$ | DCU | Read Only | No |
| **Description:** Contains the instantaneous current applied to the motor. See Appendix A for DCU (Drive Current Units). | | | | |

### 607Ch: Home Offset

| 607Ch | Home Offset | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $-2^{31} – (2^{31}-1)$ | DPU | Read / Write | Yes |
| **Description:** When the homing routine is complete, the zero position found by the drive is given an offset equal to the value stored in this object. All moves are interpreted relative to this new zero position. See Appendix A for DPU (Drive Position Units). | | | | |

### 607Ah: Target Position

| 607Ah | Target Position | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | $-2^{31} – (2^{31}-1)$ | DPU | Read / Write | No |
| **Description:** Sets the Target Position value while in profile position mode (set by object 6060h). This is the target position before limiting and profiling is applied. Position error is derived from demanded position, which is this signal after limiting and profiling is applied. See Appendix A for DPU (Drive Position Units). | | | | |

### 6086h: Motion Profile Type

| 6086h | Motion Profile Type | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer16 | $-2^{15} – (2^{15}-1)$ | N/A | Read / Write | Yes |
| **Description:** When using the profiler, (see object 6060 for setting profile modes) the value of this object determines the type of profiling the drive will perform. Presently this is a read only object and is set to perform a linear ramp when the profiler is enabled. | | | | |

### 60C1h: Interpolation Data Record

| 60C1.01h | 1st Parameter of Interpolated Function | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| PVT Data Type | N/A | N/A | N/A | Yes |
| **Description:**<br>Holds the active PVT end point. This object is not accessible. Note that DSP402 relates this object to the Interpolation Sub Mode Select object (60C0h) and reserves that object for specifying the interpolation mode the drive should use. However, AMC drives use a single interpolation method (PVT) and, as a result, there is no need for object 60C0h. | | | | |

### 60C4h: Interpolation Data Configuration

| 60C4.01h | Maximum Buffer Size | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | 15 | N/A | Read Only | Yes |
| **Description:**<br>Maximum buffer size is 15. | | | | |

| 60C4.02h | Actual Buffer Size | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | 15 | N/A | Read Only | Yes |
| **Description:**<br>Actual buffer size is 15. | | | | |

| 60C4.03h | Buffer Organization | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 | N/A | Read Only | Yes |
| **Description:**<br>Buffer is organized as FIFO (First In First Out). | | | | |

| 60C4.04h | Buffer Position | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned32 | 0-15 | N/A | Read Only | Yes |
| **Description:**<br>Number of points in the buffer. | | | | |

| 60C4.05h | Size Of Data Record | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 8 | N/A | Read Only | Yes |
| **Description:**<br>The data record is fixed to 8 bytes (PVT point length). | | | | |

| 60C4.06h | Buffer Clear | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Unsigned8 | 0 | N/A | Write Only | Yes |
| **Description:**<br>Writing a 0 (zero) to this sub-index, will clear the buffer. | | | | |

### 60FFh: Target Velocity

| 60FFh | Target Velocity | | | |
|---|---|---|---|---|
| **Data Type** | **Data Range** | **Units** | **Accessibility** | **Stored to NVM** |
| Integer32 | 0 – ($2^{32}$-1) | DSU | Read / Write | No |
| **Description:**<br>Use this object to set the Target Velocity when the drive is in Profile Velocity mode. See Appendix A for DSU (Drive Velocity Units). | | | | |

# Appendix

| Name | To Convert From | To | Multiply By | Then |
|---|---|---|---|---|
| **DAU** (Drive Accelleration Units) | Counts/$s^2$ | CANopen units | $(10^8)/(2^{32})$ | Convert to hex |
| **DCU 32bit** (Drive Current Units) | A | | 2730.6 | |
| **DCU 16bit** (Drive Current Units) | A | | 682.6 | |
| **DJU** (Drive Jerk Units) | A/s | | 17895.5 | |
| **DGU** (Drive Angle Units) | Degrees | | $2^{16}/360$ | |
| **DHU** (Drive Inductance Units) | μH | | 1 | |
| **DPU** (Drive Position Units) | Counts | | 1 | |
| **DRU** (Drive Resistance Units) | Ohms | | 1 | |
| **DSU 32bit** (Drive Velocity Units) | Counts/s | | $2^{16}/10^4$ | |
| **DSU 16bit** (Drive Velocity Units) | Counts/s | | $2^{16}/(256*10^{4)}$ | |
| **DVU** (Drive Voltage Units) | Volts | | $2^{13}/10$ | |
| **DWU** (Drive Power Units) | Watts | | 1 | |
| **DTU 32bit** (Drive Temperature Units) | C | | $2^{16}$ | |

| Name | Formula | Then |
|---|---|---|
| **DTU 16bit** (Drive Temperature Units) | CANopen units = (Degrees C *153) + 3366 | Convert to hex |

# Index

**6**



**B**



**C**



**E**



**H**



**I**



**M**



**N**



**O**

**P**



**S**



**T**