CAN ACTUATOR

1A0011GJ.doc
March 21, 2024
TH, HK, BG

INDEX

DESCRIPTION

The CAN ACTUATOR is a state of the art linear actuator with a built in microprocessor based control system. It communicates with a host with J1939 CAN at 250K or 500K baud. It is designed with CAN configurable IDs so many CAN actuators can be connected on the same CAN bus.

Each actuator consists of three main parts, the motor that moves the actuator, the clutch that allows the actuator to be disengaged, and a position/extension sensor. All three of these parts are controlled over the CAN bus. NOTE: At Power-Up, the Actuator does a self-test which moves the motor a little bit without the clutch. This should not move the shaft.

If multiple actuators are used on one CAN Bus, you need to assign each actuator a separate ID. These IDs are stored in non-volatile memory so they do not need to be reassigned after every power-up. They can be re-assigned if you wish to.

CONTROLLING POSITION

To make the actuator move to a specific position, use CAN commands to turn on the clutch then go to the required position.

DISENGAGING ACTUATOR

By turning off the clutch, the actuator's rod is free to move. This allows anything attached to the actuator to move the actuator to their resting position. For example, when an actuator is coupled to a mechanical lever, turning off the clutch allows the lever to be used manually. The minimum force required to move the shaft is about 5 pounds.

POSITION SENSOR

The internal non-contacting position sensor measures extension shaft position and can report it over the CAN bus.

INSTALLATION

Refer to actuator pictorial at the end of this manual for dimensional and mounting information. The prevent side loading the actuator shaft must be mounted straight to the equipment it is attached to. Using clevis and rod accessory will assist in alignment of the shaft and easy disassembling during set up. Remove the proper drain plug as indicated on the decals on the actuator to prevent buildup of water inside the actuator. Additionally, refer to the PINOUT sections in this manual for electrical connection.

Make sure your CAN bus has the proper terminating resistors installed. These are two 120 ohm resistors across CANH and CANL, one at either end of the CAN bus.

BEFORE APPLYING POWER!

- Check power and ground for proper polarity.

- Check the CAN wires for proper polarity.

- Read the rest of this manual.

CAN MESSAGES

IDENTIFIERS (PGNs)

The CAN actuator uses 29bit CAN 2.0b identifiers.
There are both Command Identifiers and Report Identifiers. Command Identifiers are used to send commands to the CAN actuator. Report Identifiers are used by the CAN actuator to send messages to the host.

Each CAN actuator has default Command and Report Identifiers which cannot be changed. They can be enabled and disabled if desired. At power-up, the Default Command Identifier is always enabled.

To allow for multiple CAN actuators to be used on the same CAN bus, each CAN actuator can have one User Defined Report Identifier, and up to four User Defined Command Identifiers. These are assigned using CAN messages.

The Report Identifier that the CAN actuator uses can be switched between the Default Report ID and the User Defined Report ID by clearing or setting the **RPSEL** bit in the **Report ID Reassignment** message.

All the assigned Command IDs are active at the same time. This can be helpful when setting up various groups of actuators. For example, each actuator could have one of their User Defined Command IDs and their User Defined Report IDs set up as unique to each of them. Then perhaps the second User Defined Command IDs could be set to either a Chassis ID or a Turret ID. Then a command sent from the master to the Chassis ID will be obeyed by all the actuator with this ID assigned to it. Any replies could be unique, if the modules are set to use their User Defined Report IDs by setting **RPSEL** to ‘1’.

The Default Command ID can be enabled or disabled using the **DISDEF** bit in the **Command ID Reassignment** message.

The default Command Identifier is 0xFF0000.

The default Report Identifier is 0xFF0001.

NOTE: Identifiers 0xFFFEXX and 0xFF00XX are reserved and should not be used for Used Defined IDs.

CAN COMMANDS

The following are the commands that can be sent to the actuator. These can either be sent to the Default Command Identifier, or to a User Command Identifier if it has been set up.

All messages use 8 bytes. Byte 0 is used to specify the message's Type. Byte 1 is used to flag auto replies, confirmation requests, and to further specify the message (Data Type). Bytes 2-7 are used to hold the message's data.

Byte 1		
C	A	DATA TYPE

C – Confirmation Flag:
Set to 1, the actuator will echo the same message to confirm it was received.
Set to 0, no confirmation will be sent back.

A – Auto Reply Flag:
Set to 1, the actuator will reply with a report that is set up for that message. Each command might have a different Auto Reply report. Not all messages have a corresponding Auto Reply report.
Set to 0, no reply.

Bits 0-5 are used to specify the DATA TYPE of the message.

Note: If the CAN actuator does not receive a command within 1 second, it will turn off both the clutch and the motor to go into a safe mode. It is suggested to refresh commands every 100ms or so.

Position Command

Byte 0	Byte 1			Byte 2	Byte 3			Byte 4	Byte 5	Byte 6	Byte 7
15	C	A	DT = 10	DPOS_LOW	CE	M	DPOS_HI				
0x0F	C	A	DT =0x0A	DPOS_LOW	CE	M	DPOS_HI				

This message is used to put the actuator in automatic mode where it controls the position or in passive mode where the shaft is free to move.

Byte1:

C – Confirmation Flag. (See above)

A – Auto Reply Flag. (If set, the Enhanced Position Report is replied.)

DT - Data Type - must be set to 10 (0x0A).

Byte 2:

DPOS_LOW – Desired Position – This is the position the actuator moves to automatically plus the offset of 500 (0.5”). The value is in 0.001” steps. Byte 2 is the least significant byte.

Byte 3:

CE – CLUTCH ENABLE - The most significant bit of Byte 3 engages and disengages the Clutch.

CE = 1 Clutch on.
CE = 0 Clutch off. Shaft moves freely.

M – MOTOR ENABLE - The second most significant bit of Byte 3 turns on and off the motor.

M = 1 turns on the motor and allows the actuator to move to the desired position.
M=0 turn off the motor.

DPOS_HI – Desired Position – This is the position the actuator moves to automatically plus the offset of 500 (0.5”). The value is in 0.001” steps. The lower 5 bits of Byte 3 are the most significant byte.

Byte 4 thru Byte 7:

Reserved.

Example:

(Decimal)	15	74	196	201	0	0	0	0
(HEX)	0x0F	0x4A	0xC4	0xC9	0	0	0	0

This enables the motor and clutch and moves to 2” (0x9C4 = 2500. 2500 – 500 = 2000 = 2.000”). It also asks for an Auto Reply message.

CAN ACTUATOR

NOTE: TO EXTEND THE LIFE OF THE CLUTCH, KAR-TECH SUGGESTS ENABLING THE CUTCH AT LEAST 20ms BEFORE ENABLING THE MOTOR, AND DISABLING THE MOTOR AT LEAST 20ms BEFORE DISABLING THE CLUTCH.

Example 1: Going to 2”:

Send (Just the clutch is enabled/ON.)

(Decimal)	15	74	196	137	0	0	0	0
(HEX)	0x0F	0x4A	0xC4	0x89	0	0	0	0

Wait 20ms or more

Send (Clutch and motor ON, to start motion)

(Decimal)	15	74	196	201	0	0	0	0
(HEX)	0x0F	0x4A	0xC4	0xC9	0	0	0	0

Proceed with controlled positions

Example 2: Stopping at 1.5”:

Send (Just clutch ON, to stop motion)

(Decimal)	15	74	208	135	0	0	0	0
(HEX)	0x0F	0x4A	0xD0	0x87	0	0	0	0

Wait 20ms or more

Send (Neither clutch nor motor ON)

(Decimal)	15	74	208	7	0	0	0	0
(HEX)	0x0F	0x4A	0xD0	0x07	0	0	0	0

Automatic Zero Calibration

Byte 0	Byte 1			Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
126	C	A	DT = 2	18	52	86	171	205	239
0x7E	C	A	DT =0x02	0x12	0x34	0x56	0xAB	0xCD	0xEF

This message is used to reset and calibrate the internal position sensor. With the actuator shaft disconnected mechanically from load, send this message. The actuator will fully extend and then retract itself and automatically zero its sensor. **NOTE: THIS HAS BEEN DONE IN THE FACTORY AND SHOULD NOT BE NEEDED TO BE REPEATED IN THE FIELD. ALSO, NOT DISCONNECTING THE SHAFT MAY SCREW UP THE CALIBRATION AND SUBSEQUENT MEASUREMENTS. USE WITH CAUTION.**

Byte1:

C – Confirmation Flag. (See above)

A – Auto Reply Flag. (If set, the 238 Report is replied.)

DT - Data Type - must be set to 2 (0x02).

Byte 2:

Must be set to 18 (0x12).

Byte 3:

Must be set to 52 (0x34).

Byte 4:

Must be set to 86 (0x56).

Byte 5:

Must be set to 171 (0xAB).

Byte 6:

Must be set to 205 (0xCD).

Byte 7:

Must be set to 239 (0xEF).

Example:

(Decimal)	126	2	18	52	86	171	205	239
(HEX)	0x7E	0x02	0x12	0x34	0x56	0xAB	0xCD	0xEF

This initiates the Auto Zero Calibration routine. NOTE: MAKE SURE THE ACTUATOR SHAFT IS FREE TO MOVE!

Configure Baud Rate

Byte 0	Byte 1			Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
245	C		DT = 0	0	0	BAUD RATE			
0xF5	C		DT = 0x00	0	0	BAUD RATE			

Default: 250K

The BAUD RATE is saved in the actuator and will not be lost when power is turned off.

This message is used to set the baud rate.

Byte1:

C – Confirmation Flag. (See above)

DT - Data Type - must be set to 0 (0x00).

Byte 2:

Must be set to 0.

Byte 3:

Must be set to 0.

Bytes 4:

This byte is used to configure the baud rate. Send one of the values below for desired baud rate.
0 (0x00) – 250K
1 (0x01) – 500K

Bytes 5 thru 7:

Reserved

Example:

(Decimal)	245	0	0	0	1	0	0	0
(HEX)	0xF5	0x00	0	0	0x01	0	0	0

This sets the baud rate to 500K (0x01).

Note: To change the Actuator to 500kbps, this message needs to be sent at 250kbps. Similarly, to change the Actuator to 250kbps, this message needs to be sent at 500kbps.

Report ID Reassignment

Byte 0	Byte 1			Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
247	C		DT = 0	USER DEFINED REPORT ID				RPSEL	
0xF7	C		DT = 0	USER DEFINED REPORT ID				RPSEL	

The USER DEFINED REPORT ID and RPSEL are saved in the actuator and will not be lost when power is turned off.

This message is used to configure the User Defined Report CAN ID, and to switch between the user defined one and the default report ID. **If multiple actuators are used together, make sure to assign each one a unique USER DEFINED REPORT ID so you can tell the reports apart.**

Byte1:

C – Confirmation Flag. (See above)

DT - Data Type - must be set to 0.

Bytes 2 thru 5:

USER DEFINED REPORT ID - This value is the User Defined ID that the actuator can use when sending reports. Byte 2 is the Least Significant Byte. Byte 5 is the Most Significant Byte.

NOTE: Identifiers 0xFFFEXX and 0xFF00XX are reserved and should not be used for Used Defined IDs.

NOTE: To set or clear RPSEL without affecting the previously assigned User Defined Report ID, use the Current ID or 0xFEFFFFFF.

Byte 6:

RPSEL – This parameter lets you select between either the Default Report ID or the User Defined Report ID. Set RPSEL to 1 to use the User Defined Report ID. Clear RPSEL to use the Default Report ID.

Byte 7:

Reserved.

Example:

(Decimal)	247	0	1	2	255	0	1	0
(HEX)	0xF7	0x00	0x01	0x02	0xFF	0	0x01	0

This sets USER DEFINED REPORT ID to 0x00FF0201, and selects the USER DEFINED REPORT ID as the active Report ID.

Command ID Reassignment

Byte 0	Byte 1			Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
247	C		FILT	USER DEFINED COMMAND ID					DISDEF
0xF7	C		FILT	USER DEFINED COMMAND ID					DISDEF

The USER DEFINED COMMAND IDs are saved in the actuator and will not be lost when power is turned off.

This message is used to assign User Defined Command IDs. The actuator can have up to 4 of these. Multiple command IDs makes it possible to have some commands for certain groups of actuators. For example, turning off all the "left side" actuators, all at once can be done with one CAN message if all the "left side" actuators have a common User Defined Command ID.

Byte1:

C – Confirmation Flag. (See above)

FILT – This is the index of the User Defined Command ID. The value can be between 1 and 4.

Bytes 2 thru 5:

USER DEFINED COMMAND ID - This value is the User Defined ID that the actuator can respond to. Since there can be up to 4 of these, and the default ID, the actuator can respond to up to 5 different sets of commands. Byte 2 is the Least Significant Byte. Byte 5 is the Most Significant Byte.

NOTE: Identifiers 0xFFFEXX and 0xFF00XX are reserved and should not be used for Used Defined IDs.

NOTE: To change DISDEF without changing any USER DEFINED COMMAND IDs, enter 0xFEFFFFFF in bytes 2-5.

NOTE: If priority (byte 5) is set to 0xFF all priorities can be used for user defined ID. If priority is set to anything else, the exact user defined ID must be used.

Byte 6:

Unused.

Byte 7:

DISDEF. Setting this value to 1 disables the Default Command ID.
Clearing DISDEF enables the Default Command ID.
Note: Disabling Default Command IDs can be helpful when assigning IDs to multiple actuators.

DISDEF is always cleared when the actuator is powered up, so the actuator will respond to Default Command IDs. If multiple actuators are used, keep this in mind!

Example:

(Decimal)	247	4	2	3	255	0	0	1
(HEX)	0xF7	0x04	0x02	0x03	0xFF	0	0	0x01

This sets the 4th USER DEFINED COMMAND ID to 0x00FF0302, and disables the Default Command ID. Until power is cycled, this actuator will ignore commands set to the default actuator Command ID.

Example:

(Decimal)	247	1	255	255	255	254	0	1
(HEX)	0xF7	0x01	0xFF	0xFF	0xFF	0xFE	0	0x01

This sets disables the Default Command ID, without affecting any stored USER DEFINED COMMAND IDs. Until power is cycled, this actuator will ignore commands set to the default actuator Command ID.

Reset

Byte 0	Byte 1			Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
249	C			RESET TYPE		RESET TYPE EXTENSION 1			
0xF9	C			RESET TYPE		RESET TYPE EXTENSION 1			

This message is used to reset various parts of the actuator.

Byte1:

C – Confirmation Flag. (See above)

Bytes 2 thru 3:

RESET TYPE - This value determines which type of reset is to be done.
0x0001 = Reset Outputs. All outputs are turned off.
0x0002 = Reset User Defined IDs.
0x0004 = Reset Report Rates.
0x0008 = Reset Hardware Configurations.
0x0010 = Reset HISTOGRAM.
0x0020 = Reset User Configurations (KP, KI, KD, MIN, MAX…).
0xFFFF for all four bytes (2-5) = Reset everything.
Byte 2 is the Least Significant Byte. Byte 3 is the Most Significant Byte.

Bytes 4 & 5:

RESET TYPE EXTENTION 1 – This parameter further defines some resets. Setting the bit to 1, activates the specific Reset. Clearing the bit leaves that parameter untouched.

For below parameters to be reset bytes 2-3 must be set to 2 (0x0002):
Bit 0: Reset User Defined Report ID
Bit 1: Reset User Defined Command ID #1
Bit 2: Reset User Defined Command ID #2
Bit 3: Reset User Defined Command ID #3
Bit 4: Reset User Defined Command ID #4
Bit 13: Reset RPSEL.
Bit 14: Reset DISDEF.

For below parameters to be reset bytes 2-3 must be set to 16 (0x0010):
Bit 0: Reset clutch on time in histogram
Bit 2: Reset motor on time in histogram
Bit 3: Reset total on time in histogram
Bit 4: Reset zero calibration count in histogram
Bit 5: Reset low voltage count in histogram
Bit 6: Reset circuit board over temperature duration in histogram
Bit 7: Reset clutch open load error count in histogram

Bit 8: Reset clutch overload error count in histogram
Bit 9: Reset motor open load error count in histogram
Bit 10: Reset motor overload error count in histogram

Byte 4 is the Least Significant Byte. Byte 5 is the Most Significant Byte.

Bytes 6 & 7:
Reserved.

Example:

(Decimal)	249	0	18	0	20	0	0	0
(HEX)	0xF9	0	0x12	0	0x14	0x00	0	0

This resets KP, KI, KD parameters (0x0010 for bytes 2 & 3) and resets the User Defined Command IDs #2 & #4 (0x0002 for bytes 2 & 3 and 0x0014 for bytes 4 & 5).

CAN REQUESTS

The following are the special commands that are used to request information from the actuator.

Report Polling

Byte 0	Byte 1			Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
241	C			INDEX 1	INDEX 2	INDEX 3	INDEX 4	INDEX 5	INDEX 6
0xF1	C			INDEX 1	INDEX 2	INDEX 3	INDEX 4	INDEX 5	INDEX 6

This request poles up to 6 messages/replies at once. Send this and the actuator will reply with up to 6 different messages.

Byte1:

C – Confirmation Flag. (See above)

Bytes 2 thru 7:

Each byte can be used to request a different report to be sent back in reply. Put the Index in a byte. Putting in 235 (0xFB) or higher will be ignored, so fill unused bytes with 255 (0xFF).

Example:

(Decimal)	241	0	128	129	168	255	255	255
(HEX)	0xF1	0	0x80	0x81	0xA8	0xFF	0xFF	0xFF

This requests the Position Report (0x80), the Motor Current (0x81), and the Unique Device ID Reports (0xA8).

Assigning Scheduled Report Rates

Byte 0	Byte 1			Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
240	C			INDEX 1	REPORT TIME 1 (ms)		INDEX 2	REPORT TIME 2 (ms)	
0xF0	C			INDEX 1	REPORT TIME 1 (ms)		INDEX 2	REPORT TIME 2 (ms)	

This is used to set up to 2 reports to be sent automatically at a periodic scheduled rate.

Byte1:

C – Confirmation Flag. (See above)

Byte 2:

INDEX 1: This is the Report Index that you want to start being reported periodically (scheduled).

Bytes 3 & 4:

REPORT TIME 1 - is the scheduled report period in ms reporting INDEX 1.

Byte 3 is the least significant byte. Byte 4 is the most significant byte.

Byte 5:

INDEX 2: This is the Report Index that you want to start being reported periodically (scheduled). If you do not want a second scheduled report, just set byte 5 to 235 (0xFB) or higher.

Bytes 6 & 7:

REPORT TIME 2 - is the scheduled report period in ms reporting INDEX 2.

Byte 6 is the least significant byte. Byte 7 is the most significant byte.

Example:

(Decimal)	240	0	128	100	0	255	255	255
(HEX)	0xF0	0	0x80	0x64	0	0xFF	0xFF	0xFF

This sets the Position Report to be sent once every 100ms.

Low Level Request – Software Version Data

Byte 0	Byte 1		Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
127	C	DT = 65	1					
0x7F	C	DT = 0x41	0x01					

This message is used to request the software version data.

Byte1:

C – Confirmation Flag. (See above)

DT - Data Type - must be set to 65 (0x41).

Byte2:

Must be set to 1.

Byte 3 thru 7:

Reserved.

Example:

(Decimal) 127 65 1 0 0 0 0 0
(HEX) 0x7F 0x41 0x01 0 0 0 0 0
This requests the Software Version data to be reported.

Low Level Request – Histogram 1

Byte 0	Byte 1		Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
127	C	DT = 67	1					
0x7F	C	DT = 0x43	0x01					

This message is used to request the first packet of the Histogram. In this message is total clutch on time. See Histogram 1 Report under CAN Replies.

Byte1:

C – Confirmation Flag. (See above)

DT - Data Type - must be set to 67 (0x43).

Byte2:

Must be set to 1.

Byte 3 thru 7:

Reserved.

Example:

(Decimal)	127	67	1	0	0	0	0	0
(HEX)	0x7F	0x43	0x01	0	0	0	0	0

This requests Histogram 1 to be reported.

Low Level Request – Histogram 2

Byte 0	Byte 1		Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
127	C	DT = 68	1					
0x7F	C	DT = 0x44	0x01					

This message is used to request the second packet of the Histogram. In this message is total motor on time. See Histogram 2 Report under CAN Replies.

Byte1:

C – Confirmation Flag. (See above)

DT - Data Type - must be set to 68 (0x44).

Byte2:

Must be set to 1.

Byte 3 thru 7:

Reserved.

Example:

(Decimal)	127	68	1	0	0	0	0	0
(HEX)	0x7F	0x44	0x01	0	0	0	0	0

This requests Histogram 2 to be reported.

Low Level Request – Histogram 3

Byte 0	Byte 1		Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
127	C	DT = 69	1					
0x7F	C	DT = 0x45	0x01					

This message is used to request the third packet of the Histogram. In this message is total on time of actuator, if zero calibration is occurring, and the number of times zero calibration has occurred. See Histogram 3 Report under CAN Replies.

Byte1:

C – Confirmation Flag. (See above)

DT - Data Type - must be set to 69 (0x45).

Byte2:

Must be set to 1.

Byte 3 thru 7:

Reserved.

Example:

(Decimal)	127	69	1	0	0	0	0	0
(HEX)	0x7F	0x45	0x01	0	0	0	0	0

This requests Histogram 3 to be reported.

Low Level Request – Histogram 4

Byte 0	Byte 1		Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
127	C	DT = 70	1					
0x7F	C	DT = 0x46	0x01					

This message is used to request the fourth packet of the Histogram. In this message it is shown if low voltage error is present, the number of times low voltage error has occurred, if there is an over temperature error present, and the number of times the over temperature error has occurred. NOTE: temperature is circuit board temperature not motor or clutch temperature. See Histogram 4 Report under CAN Replies.

Byte1:

C – Confirmation Flag. (See above)

DT - Data Type - must be set to 70 (0x46).

Byte2:

Must be set to 1.

Byte 3 thru 7:

Reserved.

Example:

(Decimal)	127	70	1	0	0	0	0	0
(HEX)	0x7F	0x46	0x01	0	0	0	0	0

This requests Histogram 4 to be reported

Low Level Request – Histogram 5

Byte 0	Byte 1		Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
127	C	DT = 71	1					
0x7F	C	DT = 0x47	0x01					

This message is used to request the fifth packet of the Histogram. In this message is if clutch open load error is present, the number of times the clutch open load error has occurred, if the clutch overload error is present, and the number of times clutch overload error has occurred. See Histogram 5 Report under CAN Replies.

Byte1:

C – Confirmation Flag. (See above)

DT - Data Type - must be set to 71 (0x47).

Byte2:

Must be set to 1.

Byte 3 thru 7:

Reserved.

Example:

(Decimal)	127	71	1	0	0	0	0	0
(HEX)	0x7F	0x47	0x01	0	0	0	0	0

This requests Histogram 5 to be reported

Low Level Request – Histogram 6

Byte 0	Byte 1		Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
127	C	DT = 72	1					
0x7F	C	DT = 0x48	0x01					

This message is used to request the sixth packet of the Histogram. In this message is if motor open load error is present, the number of times the motor open load error has occurred, if the motor overload error is present, and the number of times motor overload error has occurred. See Histogram 6 Report under CAN Replies.

Byte1:

C – Confirmation Flag. (See above)

DT - Data Type - must be set to 72 (0x48).

Byte2:

Must be set to 1.

Byte 3 thru 7:

Reserved.

Example:

(Decimal)	127	72	1	0	0	0	0	0
(HEX)	0x7F	0x48	0x01	0	0	0	0	0

This requests Histogram 6 to be reported

Actuator Unique Device ID Request

Byte 0	Byte 1		Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
40	C	DT = 64						
0x28	C	DT = 0x40						

This message is used to request the actuator's Unique Device ID.

Byte1:

C – Confirmation Flag. (See above)

DT - Data Type - must be set to 64 (0x40).

Byte 2 thru 7:

Reserved.

Example:

(Decimal)	40	64	0	0	0	0	0	0
(HEX)	0x28	0x40	0	0	0	0	0	0

This requests the Unique Device ID to be reported.

CAN REPLIES

The following are the report messages that the actuator can send back to the host. These can either be sent using the Default Report Identifier, or to the User Report Identifier, depending on which is selected.

Position Report

Byte 0	Byte 1			Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
128			DT = 10	SHAFT EXTENSION					
0x80			DT =0x0A	SHAFT EXTENSION					

This report is used to send the shaft extension in mills (0.001”).

Byte1:

DT - Data Type - will be set to 10 (0x0A).

Bytes 2 & 3:

SHAFT EXTENSION – This is the value of Relative Shaft Extension with an offset of 0.5” added to it. The value is in 0.001” steps. Byte 2 is the least significant byte. Byte 3 is the most significant byte.

Bytes 4 thru 7:

Reserved.

Example:

(Decimal)	128	10	202	8	0	0	0	0
(HEX)	0x80	0x0A	0xCA	0x08	0	0	0	0

This reports the Shaft Position to be 1.75”. That is 2250 (0x08CA) minus 500, all divided by 1000.

Enhanced Position Report

Byte 0	Byte 1			Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
152			DT = 0	SHAFT EXTENSION		ERRORS	MOTOR CURRENT		STATUS
0x98			DT = 0	SHAFT EXTENSION		ERRORS	MOTOR CURRENT		STATUS

This report is used to send the shaft extension as well as other information.

Byte1:

DT - Data Type - will be set to 0.

Bytes 2 & 3:

SHAFT EXTENSION – This is the value of Relative Shaft Extension.

The value is in 0.001” steps. Byte 2 is the least significant byte. Byte 3 is the most significant byte.

Byte 4:

ERRORS – This byte displays any errors or warnings in the actuator:

- Bit 0: Motor overload.
- Bit 1: Clutch overload.
- Bit 2: Motor open load.
- Bit 3: Clutch open load.
- Bit 4: Not Implemented
- Bit 5: Not Implemented
- Bit 6: Not Implemented

Bytes 5 & 6:

MOTOR CURRENT – This is the motor current.

The value is in mA. Byte 5 is the least significant byte. Byte 6 is the most significant byte.

Byte 7:

STATUS: This is reserved for internal use.

Example:

(Decimal)	152	0	178	12	8	0	0	0
(HEX)	0x98	0	0xB2	0x0C	0x08	0	0	0

This reports the Shaft Position as 2.75” (0x0CB2 = 3250. 3250 – 500 = 2750. 2750/1000 = 2.75”).There is a Clutch open Error, and the motor has stopped since there is no Motor Current.

Motor Current and Board Temperature Report

Byte 0	Byte 1				Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
129				DT = 11	MOTOR CURRENT		TEMPERATURE		TEMP STATUS	
0x81				DT =0x0B	MOTOR CURRENT		TEMPERATURE		TEMP STATUS	

This report is used to send the values of the Motor Current and the circuit board temperature.

Byte1:
DT - Data Type - will be set to 11 (0x0B).

Bytes 2 & 3:
MOTOR CURRENT – This is the motor current.
The value is in mA. Byte 2 is the least significant byte. Byte 3 is the most significant byte.

Bytes 4 & 5:
TEMPERATURE – This is the circuit board temperature.
The value is in tenths of degrees Celsius. Byte 4 is the least significant byte. Byte 5 is the most significant byte.

Byte 6:
TEMP STATUS
TEMP STATUS = 0 if positive
TEMP STATUS = 0x01 if negative
TEMP STATUS = 0x02 if uncalibrated

Byte 7:
Reserved.

Example:
(Decimal) 129 11 208 7 220 0 0 0
(HEX) 0x81 0x0B 0xD0 0x07 0xDC 0 0 0
This reports the Motor Current as 2000mA (0x07D0), and the circuit board temperature is 22.0C = 71.6F.

Software Revision Report

Byte 0	Byte 1		Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
239		DT = 65	SW VER 0	SW VER 1	SW VER 2	SW DAY	SW YEAR/MONTH	
0xEF		DT =0x41	SW VER 0	SW VER 1	SW VER 2	SW DAY	SW YEAR/MONTH	

This report shows software revision data. Current software revision is 3.06.

Byte1:
DT - Data Type - will be set to 65 (0x41).

Bytes 2 thru 4:
SW VER – These bytes show the software version.

Byte 5:
SW DAY – This is the day the software was written.

Bytes 6 & 7:
SW YEAR/MONTH - This is the month and year the software was written.

Example:

(Decimal)	239	65	3	0	6	22	107	126
(HEX)	0xEF	0x41	0x03	0	0x06	0x16	0x6B	0x7E

This reports the Software Version 3.06, day 22, month 11 (0x0B) and year 2022 (0x7E6).

Histogram 1 Report

Byte 0	Byte 1		Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
239		DT = 67	CLUTCH ON TIME LSB	CLUTCH ON TIME	CLUTCH ON TIME	CLUTCH ON TIME MSB		
0xEF		DT =0x43	CLUTCH ON TIME LSB	CLUTCH ON TIME	CLUTCH ON TIME	CLUTCH ON TIME MSB		

This report shows the first packet of the Histogram. In this message is total clutch on time.

Byte1:

DT - Data Type - will be set to 67 (0x43).

Bytes 2 thru 5:

CLUTCH ON TIME – These bytes show the total time the clutch has been on. The least significant byte is 2 and the most significant is 5. This number must be multiplied by 256 ms see example below. To reset this number see Reset command.

Byte 6 & 7:

Reserved

Example:

(Decimal) 239 67 184 76 10 0 0 0

(HEX) 0xEF 0x43 0xB8 0x4C 0x0A 0x00 0 0

This reports that the motor has been on for 2 days (0xA4CB8=675,000). (675,000*0.256s*(1day/(24*60*60s)) = 2 days)

Histogram 2 Report

Byte 0	Byte 1		Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
239		DT = 68	MOTOR ON TIME LSB	MOTOR ON TIME	MOTOR ON TIME	MOTOR ON TIME MSB		
0xEF		DT =0x44	MOTOR ON TIME LSB	MOTOR ON TIME	MOTOR ON TIME	MOTOR ON TIME MSB		

This report shows the second packet of the Histogram. In this message is total motor on time.

Byte1:

DT - Data Type - will be set to 68 (0x44).

Bytes 2 thru 5:

MOTOR ON TIME – These bytes show the total time the motor has been on. The least significant byte is 2 and the most significant is 5. This number must be multiplied by 256 ms see example below. To reset this number see Reset command.

Byte 6 & 7:

Reserved

Example:

(Decimal) 239 68 184 76 10 0 0 0

(HEX) 0xEF 0x44 0xB8 0x4C 0x0A 0x00 0 0

This reports that the motor has been on for 2 days (0xA4CB8=675,000). (675,000*0.256s*(1day/(24*60*60s)) = 2 days)

Histogram 3 Report

Byte 0	Byte 1		Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
239		DT = 69	TOTAL ON TIME LSB	TOAL ON TIME	TOTAL ON TIME	TOTAL ON TIME MSB	ZERO CAL OCCURING	ZERO CAL COUNT
0xEF		DT =0x45	TOTAL ON TIME LSB	TOAL ON TIME	TOTAL ON TIME	TOTAL ON TIME MSB	ZERO CAL OCCURING	ZERO CAL COUNT

This report shows the third packet of the Histogram. In this message is total on time of actuator, if zero calibration is occurring, and the number of times zero calibration has occurred.

Byte1:

DT - Data Type - will be set to 69 (0x45).

Bytes 2 thru 5:

TOTAL ON TIME – These bytes show the total time the actuator has been on. The least significant byte is 2 and the most significant is 5. This number must be multiplied by 256 ms see example below. To reset this number see Reset command.

Byte 6:

ZERO CAL OCCURING – This byte will show a 1 (0x01) when zero calibration is in progress. Otherwise it'll show 0 (0x00).

Byte 7:

ZERO CAL COUNT – This byte will show the number of times zero calibration has been done. To reset this number see Reset command.

Example:

(Decimal) 239 69 165 88 137 0 0 0
(HEX) 0xEF 0x45 0xA5 0x58 0x89 0x00 0x00 0x00

This reports that the motor has been on for 26.67 days = (0x8958A5=9,001,125). (9,001,125*0.256s*(1day/(24*60*60s)) = 26.67 days)

Histogram 4 Report

Byte 0	Byte 1		Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
239		DT = 70	LOW VOLTAGE ERROR	ERROR COUNT LSB	ERROR COUNT MSB	TEMP ERROR	DURATION LSB	DURATION MSB
0xEF		DT =0x46	LOW VOLTAGE ERROR	ERROR COUNT LSB	ERROR COUNT MSB	TEMP ERROR	DURATION LSB	DURATION MSB

This report shows the fourth packet of the Histogram. In this message it is shown if low voltage error is present, the number of times low voltage error has occurred, if there is an over temperature error present, and the number of times the over temperature error has occurred. NOTE: temperature is circuit board temperature not motor or clutch temperature.

Byte1:

DT - Data Type - will be set to 70 (0x46).

Byte 2:

LOW VOLTAGE ERROR – This byte will show a 1 (0x01) when input voltage is below 10.5V. Otherwise it'll show 0 (0x00).

Byte 3 & 4:

ERROR COUNT – These bytes will show the number of times the low voltage error has occurred. Byte 3 is the least significant byte and byte 4 is the most significant byte. This number must be multiplied by 256 ms see example below. To reset this number see Reset command.

Byte 5:

TEMP ERROR – This byte will show a 1 (0x01) when circuit board is over temperature (over 85 degrees Celsius). Otherwise it'll show 0 (0x00).

Byte 6 & 7:

DURATION – These bytes will show how long the board was over temperature if error had occurred. Byte 6 is the least significant byte and byte 7 is the most significant byte. To reset this number see Reset command.

Example:

(Decimal) 239 70 1 10 0 0 40 9
(HEX) 0xEF 0x46 0x01 0x0A 0x00 0x00 0x28 0x09

This reports that low voltage error is present (1, 0x01) and that it has occurred 10 (0x0A) times. It also reports that temperature error is not present (0, 0x00), but had occurred for 2344(0x0928)*0.256s = 600.064s/60s = 10.001 hours in the past.

Histogram 5 Report

Byte 0	Byte 1		Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
239		DT = 71	CLUTCH OPEN LOAD ERROR	ERROR COUNT LSB	ERROR COUNT MSB	CLUTCH OVERLOAD ERROR	ERROR COUNT LSB	ERROR COUNT MSB
0xEF		DT =0x47	CLUTCH OPEN LOAD ERROR	ERROR COUNT LSB	ERROR COUNT MSB	CLUTCH OVERLOAD ERROR	ERROR COUNT LSB	ERROR COUNT MSB

This report shows the fifth packet of the Histogram. In this message is if clutch open load error is present, the number of times the clutch open load error has occurred, if the clutch overload error is present, and the number of times clutch overload error has occurred.

Byte1:

DT - Data Type - will be set to 71 (0x47).

Byte 2:

CLUTCH OPEN LOAD ERROR – This byte will show a 1 (0x01) when no load is detected on the clutch. Otherwise it'll show 0 (0x00).

Byte 3 & 4:

ERROR COUNT – These bytes will show the number of times the clutch open load error has occurred. Byte 3 is the least significant byte and byte 4 is the most significant byte. To reset this number see Reset command.

Byte 5:

CLUTCH OVERLOAD ERROR – This byte will show a 1 (0x01) when clutch overload error is present. Otherwise it'll show 0 (0x00).

Byte 6 & 7:

ERROR COUNT – These bytes will show the number of times the clutch overload error has occurred. Byte 6 is the least significant byte and byte 7 is the most significant byte. To reset this number see Reset command.

Example:

(Decimal) 239 71 1 12 0 0 0 0
(HEX) 0xEF 0x47 0x01 0x0C 0x00 0x00 0x00 0x00

This reports that clutch open load error is present (1, 0x01) and that it has occurred 12 (0x0C) times. It also reports that clutch overload error is not present (0, 0x00) and has never occurred (0, 0x00).

Histogram 6 Report

Byte 0	Byte 1		Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
239		DT = 72	MOTOR OPEN LOAD ERROR	ERROR COUNT LSB	ERROR COUNT MSB	MOTOR OVERLOAD ERROR	ERROR COUNT LSB	ERROR COUNT MSB
0xEF		DT =0x48	MOTOR OPEN LOAD ERROR	ERROR COUNT LSB	ERROR COUNT MSB	MOTOR OVERLOAD ERROR	ERROR COUNT LSB	ERROR COUNT MSB

This report shows the sixth packet of the Histogram. In this message is if motor open load error is present, the number of times the motor open load error has occurred, if the motor overload error is present, and the number of times motor overload error has occurred.

Byte1:

DT - Data Type - will be set to 72 (0x48).

Byte 2:

MOTOR OPEN LOAD ERROR – This byte will show a 1 (0x01) when no load is detected on the motor. Otherwise it'll show 0 (0x00).

Byte 3 & 4:

ERROR COUNT – These bytes will show the number of times the motor open load error has occurred. Byte 3 is the least significant byte and byte 4 is the most significant byte. To reset this number see Reset command.

Byte 5:

MOTOR OVERLOAD ERROR – This byte will show a 1 (0x01) when motor overload error is present. Otherwise it'll show 0 (0x00).

Byte 6 & 7:

ERROR COUNT – These bytes will show the number of times the motor overload error has occurred. Byte 6 is the least significant byte and byte 7 is the most significant byte. To reset this number see Reset command.

Example:

(Decimal)	239	72	0	0	1	0	44	1
(HEX)	0xEF	0x48	0x00	0x00	0x01	0x00	0x2C	0x01

This reports that motor open load error is not present (0, 0x00) and has never occurred (0, 0x00). It also reports that motor overload error is present (1, 0x01) and has occurred 300 (0x012C) times.

Unique Device ID Report Message 1

Byte 0	Byte 1			Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
168		A	DT = 0	ACTUATOR ID PART 1					
0xA8		A	DT = 0	ACTUATOR ID PART 1					

This report shows the first half of a unique 12 byte Device ID for this actuator. It is intended to be used to help differentiate actuators in a multiple actuator system. This report cannot be scheduled.

Byte1:

A – Auto Reply Flag. If this is the response from an Auto Reply, this bit will be set. Otherwise it will be cleared.

DT - Data Type - will be set to 0.

Bytes 2 thru 7:

ACTUATOR ID PART 1– These bytes show the first 6 bytes of the Unique ID Number.

Example:

(Decimal)	168	0	18	52	86	1	170	204
(HEX)	0xA8	0	0x12	0x34	0x56	0x01	0xAA	0xCC

This reports the first 6 bytes of the Unique Device ID as 0xCCAA01563412.

Unique Device ID Report Message 2

Byte 0	Byte 1			Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
168		A	DT = 1	ACTUATOR ID PART 2					
0xA8		A	DT =0x01	ACTUATOR ID PART 2					

This Report shows the second half of a unique 12 byte Device ID for this actuator. It is intended to be used to help differentiate actuators in a multiple actuator system. This Report cannot be scheduled.

Byte1:

A – Auto Reply Flag. If this is the response from an Auto Reply, this bit will be set. Otherwise it will be cleared.

DT - Data Type - will be set to 1 (0x01).

Bytes 2 thru 7:

ACTUATOR ID PART 2– These bytes show the last 6 bytes of the Unique ID Number.

Example:

(Decimal)	168	1	18	52	86	1	170	204
(HEX)	0xA8	1	0x12	0x34	0x56	0x01	0xAA	0xCC

This reports the second 6 bytes of the Unique Device ID as 0xCCAA01563412.

Zeroing Message

Byte 0	Byte 1		Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
238		66	CHIP 1 DATA		CHIP 2 DATA		ERROR 1	ERROR 2
0xEE		0x42	CHIP 1 DATA		CHIP 2 DATA		ERROR 1	ERROR 2

This report is only in response to a zeroing command. It cannot be otherwise polled or scheduled.

Byte1:
DT - Data Type – This will be a 66 (0x42) in response to an Automatic Zeroing Command, if the Auto Reply bit was set.

Bytes 2 & 3:
CHIP 1 DATA – These bytes show the data on chip 1 in bits.
Byte 2 is the least significant byte. Byte 3 is the most significant byte.

Bytes 4 & 5:
CHIP 2 DATA – These bytes show the data on chip 2 in bits.
Byte 4 is the least significant byte. Byte 5 is the most significant byte.

Byte 6:
CHIP ERROR 1 – This byte shows the error on chip 1.

Byte 7:
CHIP ERROR 2 – This byte shows the error on chip 2.

Example:

(Decimal)	238	66	232	3	86	1	170	204
(HEX)	0xEE	0x42	0xE8	0x03	0x56	0x01	0xAA	0xCC

This reports the CHIP 1 DATA as 1000 (0x03E8), CHIP 2 DATA as 342 (0x0156), CHIP ERROR 1 as 170 (0xAA) and CHIP ERROR 2 as 204 (0xCC).

WIRING

Connector: Deutsch DT04-4P-E008

PIN	DESCRIPTION
1	POWER (10-30)
2	GROUND
3	CAN HIGH
4	CAN LOW

TROUBLESHOOTING

This next section provides basic operator level troubleshooting for the CAN ACTUATOR If, after following these instructions, the system still does not function, contact your KAR-TECH representative for further instructions or servicing.

TROUBLESHOOTING CHART

PROBLEM	SOLUTION
Actuator Shaft Doesn't Move	1. With power off, check to make sure the shaft moves freely and there is no binding when attached to equipment. 2. Check that the maximum mechanical load does not exceed actuator capacity 3. Check that actuator power is on 4. Check that CAN is working at 250K Baud 5. Check that the CAN terminating resistors are installed and correct value. 6. Check that CAN ID is set correctly

Position Reading Not Correct	1. Check that actuator power is on 2. Check that CAN is working 3. Check that CAN ID is set correctly 4. If everything above looks good, disconnect the actuator shaft mechanically from anything, and perform an Automatic Zero Calibration (page 8).
HW Warnings reported in Enhanced Position Report	1. Typically this only occurs if the actuator was disassembled and then reassembled incorrectly. Disconnect the actuator shaft mechanically from anything, and perform an Automatic Zero Calibration (page 8). 2. If that fails, return the actuator to the manufacturer.

PARTS LIST

PART NUMBER	DESCRIPTION
1A0014G	3” LINEAR CAN ACTUATOR
1A0018A	ACTUATOR LINKAGE KIT

There are no user-serviceable parts inside the actuator. Return the units for service.

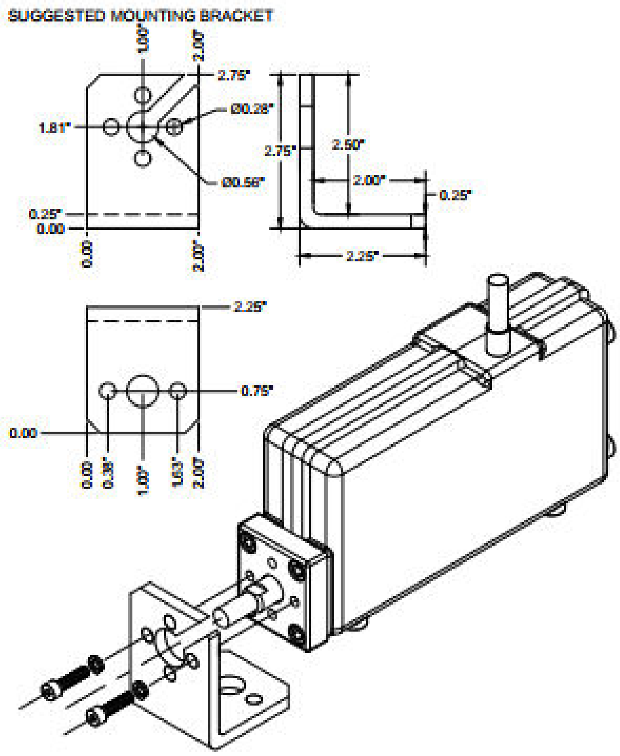
Note: For operation with negative ground systems only.

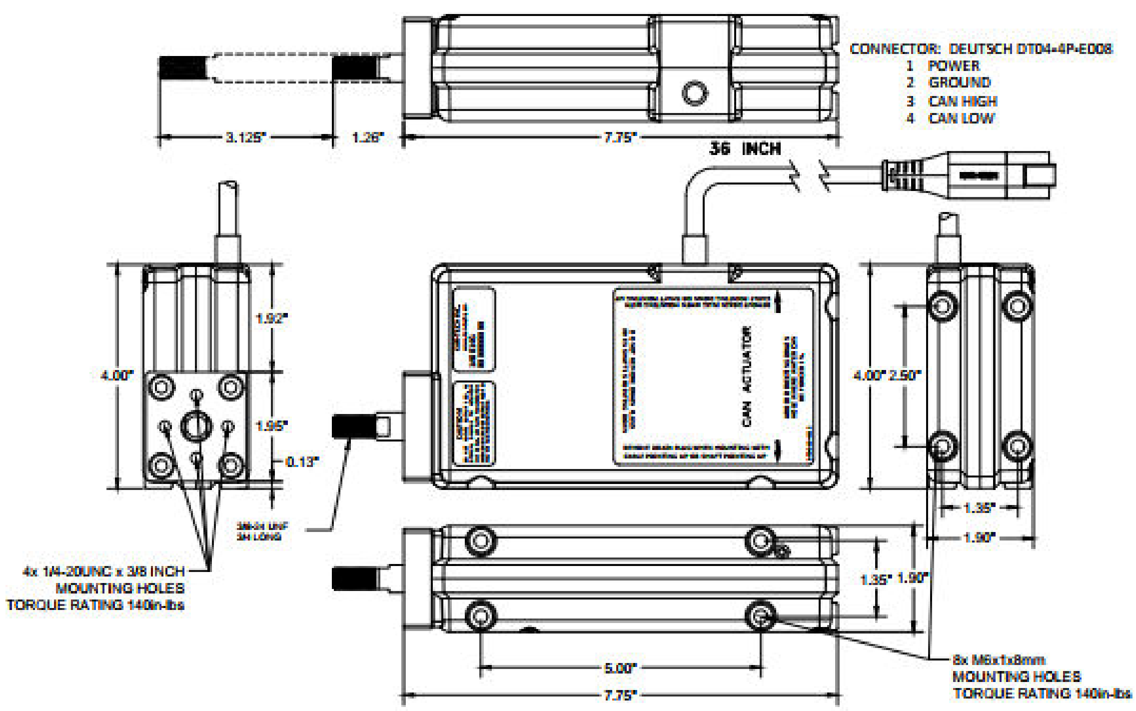
WARNING:

The CAN ACTUATOR must be operated in compliance with all applicable safety regulations, rules, and practices. Failure to follow required safety practices may result in death or serious injury.

The information, specifications, and illustrations in this manual are those in effect at the time of printing. We reserve the right to change specifications or design at any time without notice.

ACTUATOR PICTORIAL





SPECIFICATIONS

Power supply voltage.. 10-30VDC

Standby current .... 30 mA

Peak current 16A

Max current 7A

Actuator working distance.. 3.1 Inches

CAN baud rate....................................... 250K (default) or 500K

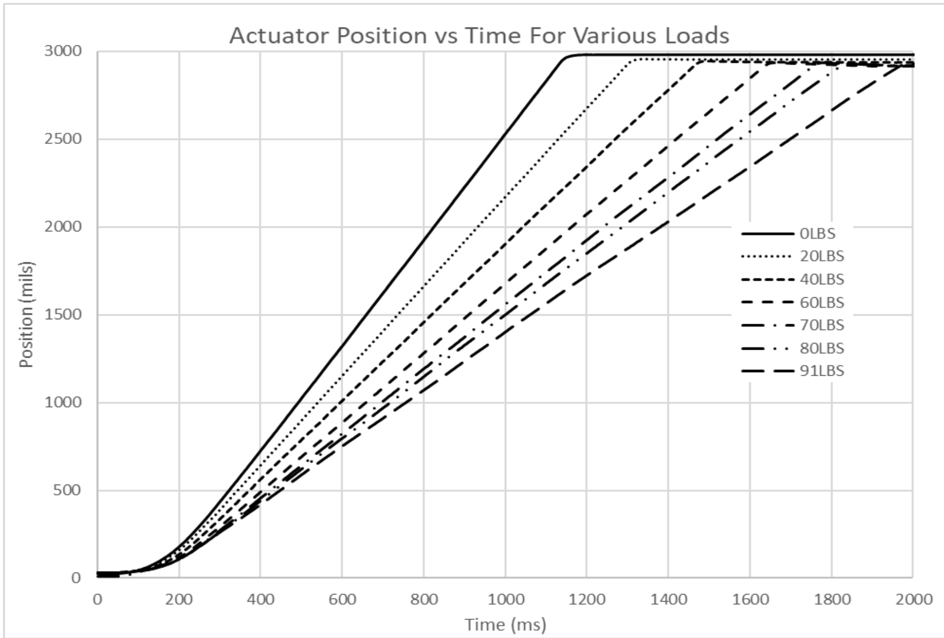
Operating temperature..................................... -40˚C to +85˚C

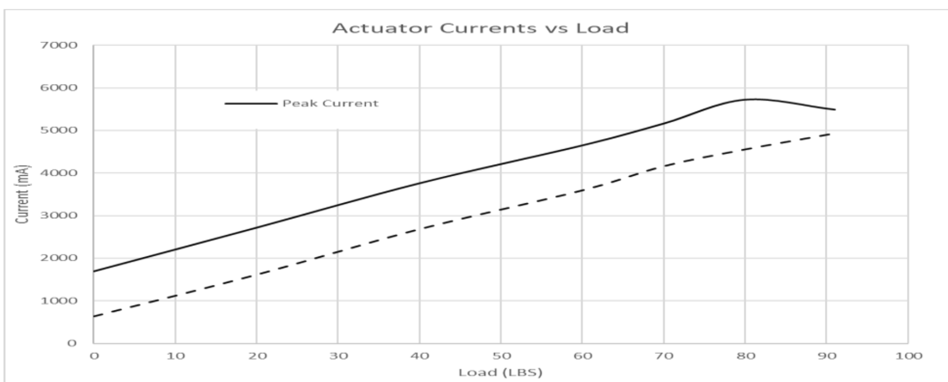
Storage temperature....................................... -40˚C to +100˚C

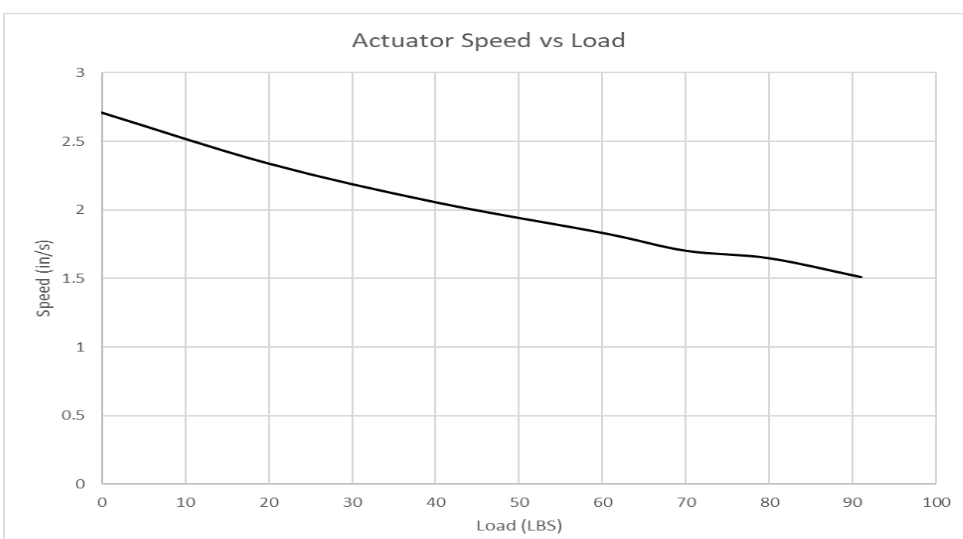
Ingress Protection.. IP 66

CAN ACTUATOR

TEST RESULTS







CAN Actuator Frequently Asked Questions:

IDs:

Q:

We understand that every power-up the default Command ID and Report ID is taken.
In my case I am doing power up of 2 CAN actuators via single CAN message at same time, how to configure user defined Command ID and Report ID differently for each actuator.

A:

Command ID:

- At power-up, the Default Command ID is enabled. However, all of the other User Defined Command IDs, if assigned, are also enabled. You can have up to 4 of these on each Actuator.
- User Defined Command IDs are always enabled.
- Assign one User Defined ID to one actuator, and a different User Defined Command ID to the other actuator. The easiest way to do this is to plug in only one actuator at a time, use the Default Command ID to do the Command ID Reassignment command (page 15)
- Once each actuator has its own User Defined Command ID, only use those User Defined Command IDs from that point on.

Report ID:

- Assign each actuator with its own User Defined Report ID (page 14). Again it is easiest to do this with one actuator plugged in at a time, and use the Default Command ID to send the Report ID Reassignment.
- Set the RPSEL bit in byte 6 of that Report ID Reassignment command. This makes the actuator reply using this User Defined Report ID and not the Default Report ID.

Note: The default Command ID is always enabled after power-up in order to enable reconfiguring, in case the programmed IDs are forgotten. For systems with more than one actuator, the User Defined Command and Report IDs should be used for normal operations.

Q:

Is the Report Identifier basically the CAN ID of the actuator?

A:

The Report ID is the CAN ID the actuator uses to send out information (reports). There is also a Command ID (that should be different than the Command ID) that the master uses to send commands to the actuator.

Q:

If we performed a Command ID Reassignment and wanted our message to be FF01 from source address 2 (i.e. 18FF0102h), not ignoring the default command ID and allowing any priority level, would this be the properly formatted message to send using the default Command Identifier 0xFF0000 (i.e. 18FF0000h)?

F7 01 02 01 FF FF 00 00

A:

With our older code, we had a bug where we were filtering for priority as well. What they have listed is correct now. If they have an old actuator, they might need to include the priority they intend to use, like F7 01 02 01 FF 18 00 00

Motion & Movement:

Q:

Can you confirm the sequence below as the proper way to execute a movement? Is there a simpler way to perform a movement? We have a dial that generates a position value via J1939 and we want to use the dial to scale between zero and a set maximum extension (somewhere around 1"), and this seems like a complicated and clunky way to execute the move.

SEQUENCE (target position is 1", no reply)
(target position set, clutch enabled, motor off)
0F 1A FC 85 00 00 00 00
20ms pause
(target position set, clutch enabled, motor on)
0F 1A FC C5 00 00 00 00
{{How much of a pause should there be to fully execute the move before the motor is turned back off?}}
(target position set, clutch enabled, motor off)
0F 1A FC 85 00 00 00 00

POSITIONAL VALUES
500, or 0” extended – 1F4
1500, or 1” extended – 5FC

BYTE BREAKDOWN
0F – Message type identifier
1A (echo, no reply)
FC – partial desired position
Byte 3 options:
05 – partial desired position, clutch off, motor off
85 – partial desired position, clutch on, motor off
C5 – partial desired position, clutch on, motor on

A:

Byte 1 should be 0x0A if no reply is desired and 0x4A if a reply is desired. The time depends on how much load is on the actuator. The last page of the datasheet has typical speeds for different loads. If you are moving only 1”, you should be able to move that far in less than 1s. Keep in mind that if you need to move further, like 3”, one command might not be enough to move all the way. The Kar-Tech CAN Actuator needs a command every 1s or less to keep the clutch activated and the position command valid. This is a failsafe feature so that if there is a CAN failure, the actuator lets go of the clutch and goes idle.

To extend the life of the clutch, especially with heavy loads, to go to 1”, use a sequence like:

0F 0A FC 85 00 00 00 00
(20ms)
0F 0A FC C5 00 00 00 00 (Repeat every 900ms if you want the clutch to stay on)
(900ms)
0F 0A FC 85 00 00 00 00
(20ms)
0F 0A FC 05 00 00 00 00

You could also enable the automatic reply and when you see the end position is reached, then turn off the motor and clutch:

0F 0A FC 85 00 00 00 00
(20ms)
0F 4A FC C5 00 00 00 00 (Repeat every 100ms until you see the position is at 1”, or within a deadband of 1”)

0F 0A FC 85 00 00 00 00
(20ms)
0F 0A FC 05 00 00 00 00

Q:

After we have command the actuator beyond 600 on a previous move, however, it quickly fails to respond to commands to move much further back, often stopping between a reported position of 565 and 595 when commanded to 500. Sending lower commands both via software and manually, such as 400 or even 0, similarly have zero effect and the actuator fails to respond of update position.

A:

First, looking at the command structure, the value is offset by 500. This makes a command of 500 is the same as 0”, or fully retracted. Any commands of 500 or less will be ignored.

To protect the actuator from slamming into the limits at 0” or 3” we added a small 0.05” protective band at each end of motion. So the working range is 0.05” to 2.95” (550 to 3450). Any commands outside this range will be ignored.

We suggest they change their coupling to have their Home position further out, for example at 0.1” (600).