2A Stepper Drive, CAN J1939

with Additional I/O

<u>USER MANUAL</u>

P/N: AX100700

VERSION HISTORY

Version	Date	Author	Modifications
1.0.0	June 15, 2010	Anna Murray	Initial Draft
1.1.0	August 19, 2010	Anna Murray	Minor updates and edits. Added clarification about the default settings versus received commands.
2.0.0	November 1, 2010	Anna Murray	Minor corrections
2.1.0	May 16, 2012	Anna Murray	Added section explaining how to reflash the ECU via CAN with the Axiomatic EA Bootloader. Updated the PID function block with new setpoints. Changed "Rotation Angle per Step" from a WORD to a FLOAT.
2.2.0	September 1, 2023	Kiril Mojsov	Performed Legacy Updates

ACCRONYMS

ACK   Positive Acknowledgement   (from SAE J1939 standard)

ADC   Analog-to-Digital Converter, 12-bit processor peripheral

AIN   Analog Input used to measure Voltage or Current signals

DIN   Digital Input used to measure active high or low signals

DM   Diagnostic Message   (from SAE J1939 standard)

DOUT   Digital Output, sourcing (high-side) output up to 5A current

DTC   Diagnostic Trouble Code   (from SAE J1939 standard)

EA   The Axiomatic Electronic Assistant (A Service Tool for Axiomatic ECUs)

ECU   Electronic Control Unit   (from SAE J1939 standard)

FIN   Frequency Input used to measure Frequency, RPM or PWM signals

I/O   Inputs and Outputs

NAK   Negative Acknowledgement   (from SAE J1939 standard)

PDU1   A format for messages that are to be sent to a destination address, either specific or global   (from SAE J1939 standard)

PDU2   A format used to send information that has been labeled using the Group Extension technique, and does not contain a destination address.

PGN   Parameter Group Number   (from SAE J1939 standard)

PropA   Message that uses the Proprietary A PGN for peer-to-peer communication

PropB   Message that uses a Proprietary B PGN for broadcast communication

PWM   Pulse Width Modulation

RMS   Root Mean Squared

RPM   Rotations per Minute

SPN   Suspect Parameter Number   (from SAE J1939 standard)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

REFERENCES

A3979	26184.23E, DMOS Microstepping Driver with Translator, Allegro Microsystems Inc.
J1939	Recommended Practice for a Serial Control and Communications Vehicle Network, SAE, October 2007
J1939/21	Data Link Layer, SAE, December 2006
J1939/71	Vehicle Application Layer, SAE, January 2009
J1939/73	Application Layer-Diagnostics, SAE, September 2006
J1939/81	Network Management, SAE, May 2003
UMAX07050x	User Manual, Axiomatic Electronic Assistant and USB-CAN, Axiomatic Technologies, 2023

> NOTE: When a description is in "**double-quotes**" and bolded, this refers to the name of a user configurable setpoint (variable). If it is in '*single-quotes*' and italicized, it refers to an option for the associated setpoint.
>
> For example: "**Default Step Type**" set to *'SixteenStep'*

This document assumes the reader is familiar with the SAE J1939 standard. Terminology from the standard is used, but is not described in this document.

1. OVERVIEW OF CONTROLLER

1.1. Description of Motor Drive

The Stepper Motor Drive is designed to give the user full access to the 2A drive control chip A3979 via J1939 command messages. It allows for simple and accurate control of the levels, waveforms, timing, and phase-shifts of the current for a two phase stepper motor.

The drive allows the user to set the default values for the stepper motor using the Axiomatic Electronic Assistant. It also allows the user to set the default targets and ranges for the run-time variables such as the step rate the motor is to run at, or the number of steps the motor is to take before holding.

A unique feature of the drive is that not only are the run-time variables accessible through the Axiomatic EA, but they can also be independently controlled by J1939 messages received while the unit is operational. This means that while the Axiomatic EA tool is used for initial setup and configuration, all the drive values can be changed as needed without it.

The default configuration for the drive is to be a purely reactive control based on the command data received from J1939 messages. However, it does have more sophisticated logic supported for more complex controls. In addition to the logic described in this user manual, other responses or function blocks could be added to the drive on request. For more information about this option, contact Application Engineering at support@axiomatic.com.

The "off-the-shelf" logic blocks supported by the drive, and selectable by the user, are:
a) Automatic PID control of the direction and step rate when the *"Motor Mode"* is selected to be controlled by the "PID Control Function" block.
b) Diagnostic detection, reaction and feedback
c) Up to five acceleration/deceleration profiles that can be setup using the Axiomatic EA, then selected during operation via a CAN message.
d) Automatic resetting of the "Number of Steps" command so that the controller can simply send the same CAN message again to step the same number of steps.

The drive also has two inputs that could be used as the command source for a run-time parameter, rather than a CAN message. Alternatively, the inputs could be used for feedback in a PID control, or as simple digital controls, for example to E-Stop the device. Lastly, they could independently be sent to the J1939 network as input data for another device on the CAN bus.

Lastly, the drive has a single digital output, capable of sinking up to 5A, to drive a relay, brake or other load in a system. By default, the output is controlled by a CAN message received from the J1939 network, but it too could be controlled by one of the inputs instead. The output can be setup for a simple on/off response, or to blink indicating a fault has been detected by the drive.

To summarize, the stepper drive is a highly programmable controller, allowing the user to configure it for their application. Its sophisticated control algorithms allow for open or closed loop drive of the motor. All I/O on the unit are inherently independent from one another, but can be programmed to interact in a large number of ways.
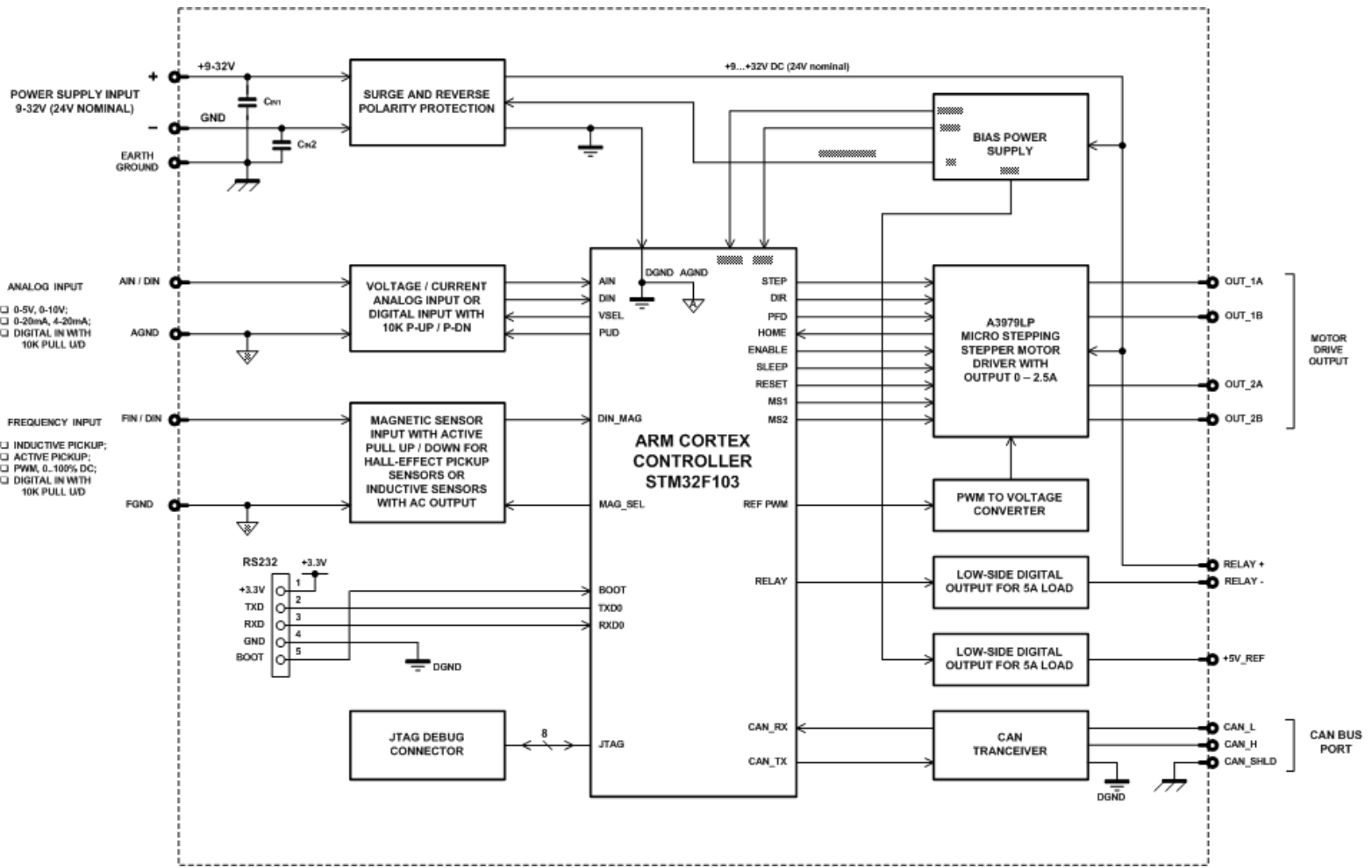
Figure 1 – 2A Stepper Motor Drive Block Diagram

The various function blocks supported by the drive are outlined in the following sections. All setpoints are user configurable using the Axiomatic Electronic Assistant, as outlined in Section 4 of this document.

- Yellow bubbles represent setpoints that are unique to the function block in question

- Green bubbles represent the inputs to the function block

- Grey (light purple) bubbles represent outputs from the function block that can be mapped as inputs to other function blocks

- Blue bubbles represent another function block supported by the ECU which is intrinsically linked to the function block in question.

- Red bubbles represent outputs from the function block that are used internally in the controller by other function blocks, but are not accessible to the user.

- Dark purple bubbles represent messages that are sent to or received from the CAN bus.

1.2. Motor Drive Function Blocks

There are two type of function blocks associated with the stepper drive, "Values" and "Controls". The "Values" function block contains the default drive setpoints, and the limits for the linear controls. The "Controls" function block tells the drive what it should use as the source of the command signal for each of the tasks that it supports.
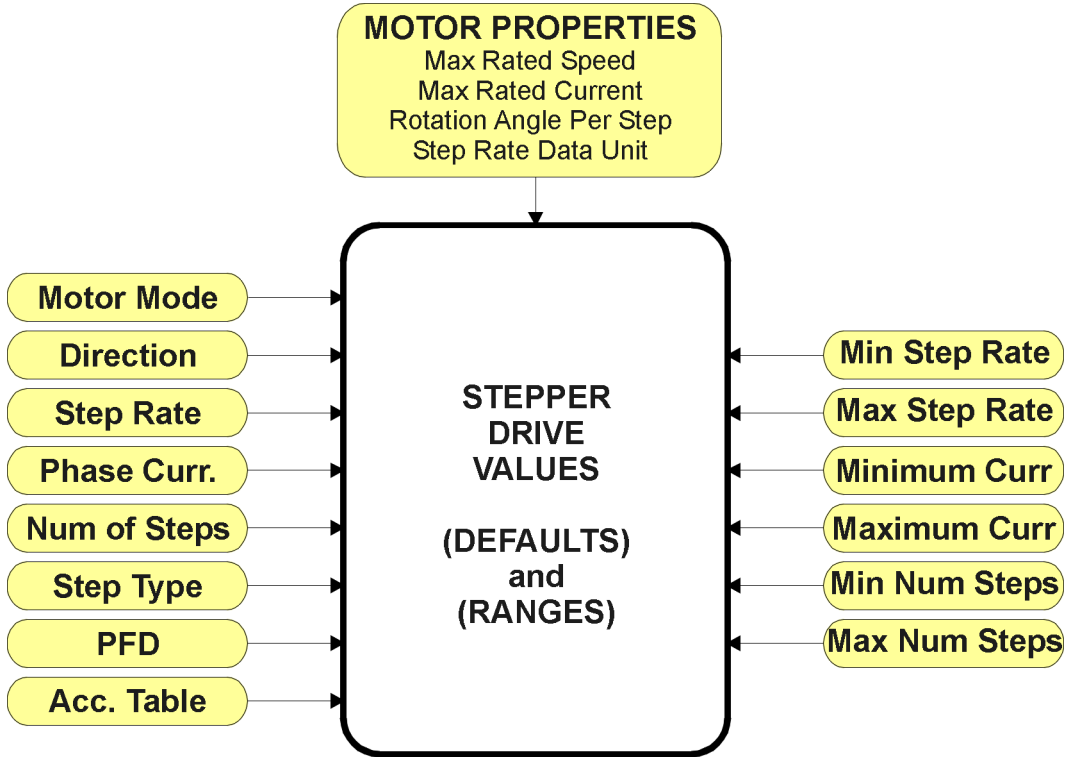
Figure 2 – Stepper Drive Values Function Block
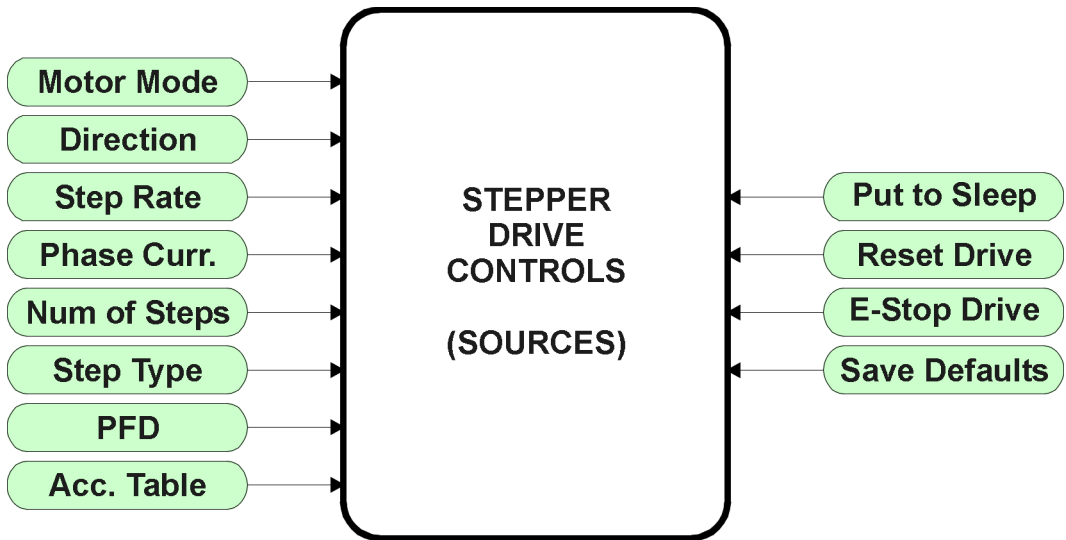
Figure 3 – Stepper Drive Controls Function Block

Unless noted otherwise, all drive functions described in the subsequent 1.2.x sections are available both through the Axiomatic EA, and as read from a CAN message (or other control source).

1.2.1. Rotation Angle Per Step

This setpoint is only accessible through the Axiomatic EA in the "Stepper Drive Values" function block. It is to be set by the user to reflect the stepper motor's step angle. This value is used by the drive when the step rate sent to the unit is in RPM. It uses it to calculate the corresponding Step/Sec signal to apply to the drive chip. By default, this value is set to 18º.

1.2.2. Step Rate Data Unit

This setpoint is only accessible through the EA in the "Stepper Drive Values" function block. It is used by the driver to interpret the data sent by the Step Rate control. By default, step rate data is read as *'Steps Per Second,'* and the available options for this setpoint are listed in the following table.

0	*Steps Per Second*
1	*Rotations Per Minute*

Table 1 – Step Rate Data Unit Options

1.2.3. Motor Mode

This is the global control function for the drive. As a CAN command, it can be sent as one of three states, *'Disabled', 'Stopped'* or *'Running'.* As a digital input, it will either be commanded to the *'Stopped'* (input OFF) or *'Running'* (input ON) modes.

As a feedback signal, however, there are two additional modes that will be set by the drive, *'Stepping'* and *'Holding'*. These modes reflect how the drive is actually operating.

"Default Motor Mode" is a read-only setpoint, as the unit always starts up in *'Stopped'* mode after a power-up or reset. The available options for this setpoint are listed in the following table.

0	*Disabled*
1	*Stopped*
2	*Running*
3	*Stepping (read-only)*
4	*Holding (read-only)*

Table 2– Motor Mode Options

When commanded to go to the *'Disabled'* mode, the drive will immediately shutoff (no current through the phases) and reset.

When no acceleration table has been selected, and when the number of steps has been commanded to zero (continuous operation when running), the *'Stopped'* mode behaves the same as *'Disabled.'*

Motor Mode Used with an Acceleration Table

Should an acceleration table be engaged, when commanded to a *'Stopped'* mode while running or stepping, the motor will **decelerate** to zero from whatever step rate it was applying at the time the *'Stopped'* command was received. (Note: This does not apply if the unit is in a *'Holding'* mode, it will simply shutoff) More information about acceleration/deceleration profiles, refer to section 1.2.10.

If the motor has been *'Stopped'* (not holding) and is then commanded to *'Running'* while an acceleration table has been selected, the motor will **accelerate** from zero to the commanded step rate. If no acceleration table is used, then the motor will immediately start stepping at the commanded rate as soon as the motor mode is commanded to *'Running.'*

Motor Mode Used with Number of Steps

When the unit is commanded to the *'Running'* mode, it will drive the motor the commanded **"Number of Steps."** If this is a non-zero value, it step the motor X steps, and will report its status as *'Stepping'* in a feedback message. Once it has reached X steps, it will stop spinning the motor, and hold the phases at the current level it was at when it finished stepping. In this case, it will report its status as *'Holding.'*

If a zero value has been commanded for the **"Number of Steps,"** the drive will run the motor continuously at the commanded **"Step Rate"** while the commanded **"Motor Mode"** remains as *'Running.'*

If the CAN message commanding the **"Number of Steps"** control is expected to be sent repeatedly (i.e. the **"Received Message Timeout"** is a non-zero value), then the **"Motor Mode"** signal can be used to re-activate the stepping action.

To clarify, every time the **"Number of Steps"** command changes in the CAN message, the unit will step the motor X steps. However, should it be desired to step again the same number of steps as used the last time the motor moved, the drive will not respond to the same data sent over and over, as it only reacts to changing data.

To work around this condition, when the CAN message is repeatedly broadcasted at some regular time interval, the **"Motor Mode"** data can be toggled from 1=Stopped to 2=Running and back again to trigger a new action to move X steps. Every time the mode changes, the drive will go into *'Stepping'* mode for X steps, then revert back to *'Holding'* mode. To shut off the drive completely in this case, the mode would have to be commanded to *'Disabled.'*

Alternatively, the **"Number of Steps"** command can be sent only once each time the drive is to step the motor. In this case, the **"Received Message Timeout"** will be set to zero, and after the drive has finished stepping the motor, it will reset the CAN message data to "not available" such that the unit will react to the same data the next time it receives the CAN message.

In this case, the toggling action of the **"Motor Mode"** command is not required. Therefore, this control is used to establish how the drive will react after the stepping has completed. In *'Stopped'* mode, it returns to zero current through both phases. In *'Running'* mode it will go to the *'Holding'* mode after it has finished stepping.

Motor Mode Used with PID Control

The last point to mention about the **"Motor Mode"** is that it is the primary control source to change to operate the drive with the PID control function. When the **"Control Source, Mode Mode"** setpoint is set to *'PID Function Block Control',* the drive will automatically update the **"Control Source, Direction"** and **"Control Source, Step Rate"** setpoints to also use the PID control.

When the PID control is used, the motor's mode, direction and speed are all controlled via the PID algorithm selected. All other parameters such as phase current or step type will run using their default value, or can still be changed via CAN messages. For more information about the PID control, refer to section 1.3

1.2.4. Direction

This setpoint is used by the driver to control the direction of rotation of the motor. By default, the motor will spin in the *'ClockWise'* direction, and the available options for this setpoint are listed in the following table.

0	CounterCW
1	ClockWise

Table 3 – Motor Direction Options

1.2.5. Step Rate

This setpoint is used by the driver to control the speed of the motor. It is a linear control, and therefore the data that is used to control the speed does not necessarily have to be in the same units as the speed data.

> ℹ️ In the **"Step Rate"** setpoints, a *'Step'* is interpreted as the full 360° cycle as shown in Figures 5 and 6. (i.e. from Home position to Home position)

For example, a CAN message could be sent as a 0 to 100% value and the step rate will be calculated according to the formula shown in Figure 4. In this case, the Xmin and Xmax values would be 0 and 100, as set in **"Received Data Min"** and **"Received Data Max",** while the Ymin and Ymax would be setpoints **"Limit Step Rate Min"** and **"Limit Step Rate Max"** respectively.

$$y = mx + a$$

$$m = \frac{Y\max - Y\min}{X\max - X\min}$$

$$a = Y\min - m * X\min$$

Figure 4 – Linear Slope Calculations

Depending of the value of **"Step Rate Data Unit"**, the drive will interpret the calculated 'Y' value as either [Step/s] or [RPM]. It would then set the appropriate rate for the drive signal to the chip controlling the motor phases.

Therefore, all the user has to do is set the minimum and maximum step rates according to the stepper motor specifications and their application. Defaults and ranges for the step rate setpoints are listed in section 4.1.

1.2.6. Max. Phase Current

This setpoint is used by the driver to control the amplitude of the phase current at the maximum point in the step. Refer to the figures in section 1.2.8 for more information. Please note, this value does NOT reflect the average RMS current through each phase.

This setpoint is also a linear control. Like the step rate, the maximum phase current set for the driver chip will be calculated using the formula in Figure 4. The X axis is represented by the received CAN message setpoints **"Received Data Min"** and **"Received Data Max"** while the Y axis is calculated based on the setpoints **"Limit Phase Current Min"** and **"Limit Phase Current Max."**

Note, however, that Ohm's law will still apply. If the calculated phase current setting is higher than what the stepper motor is capable of drawing at the applied voltage level, then the motor will simply draw whatever it is capable of.

Therefore, the user should set the minimum and maximum currents according to the stepper motor specifications and their application. Defaults and ranges for the phase current setpoints are listed in section 4.1.

1.2.7. Number of Steps

This setpoint is used by the driver step the motor a set number of steps, and then either stop or hold it. Please refer to section 1.2.3 for more information about how to use this control in conjunction with the motor mode.

This setpoint is also a linear control. Like the step rate, the number of steps will be calculated using the formula in Figure 4. The X axis is represented by the received CAN message setpoints **"Received Data Min"** and **"Received Data Max"** while the Y axis is calculated based on the setpoints **"Limit Number of Steps Min"** and **"Limit Number of Steps Max."**

"Limit Number of Steps Min" is a read-only setpoint, and is set to zero representing continuous operation. The user should set the maximum according to their application. Defaults and ranges for the number of steps setpoints are listed in section 4.1.

> *In the **"Number of Steps"** setpoints, a 'Step' is interpreted as a 90°shift of the rotor with respect to the driving phases. There are always 4 steps per 360°cycle as shown in Figures 5 and 6.*

1.2.8. Step Type

This setpoint is used by the driver to control the microsteps used to get different phase current waveforms. By default, the step type used is *'SixteenStep'*, and the available options for this setpoint, as supported by the driver chip A3979, are listed in the following table.

0	*FullStep*
1	*HalfStep*
2	*QuarterStep*
3	*SixteenStep*

Table 4 – Step Type Options

The following figures, extracted from the Allegro datasheet for the driver chip, show the current waveforms on each phase depending on the step type selected.
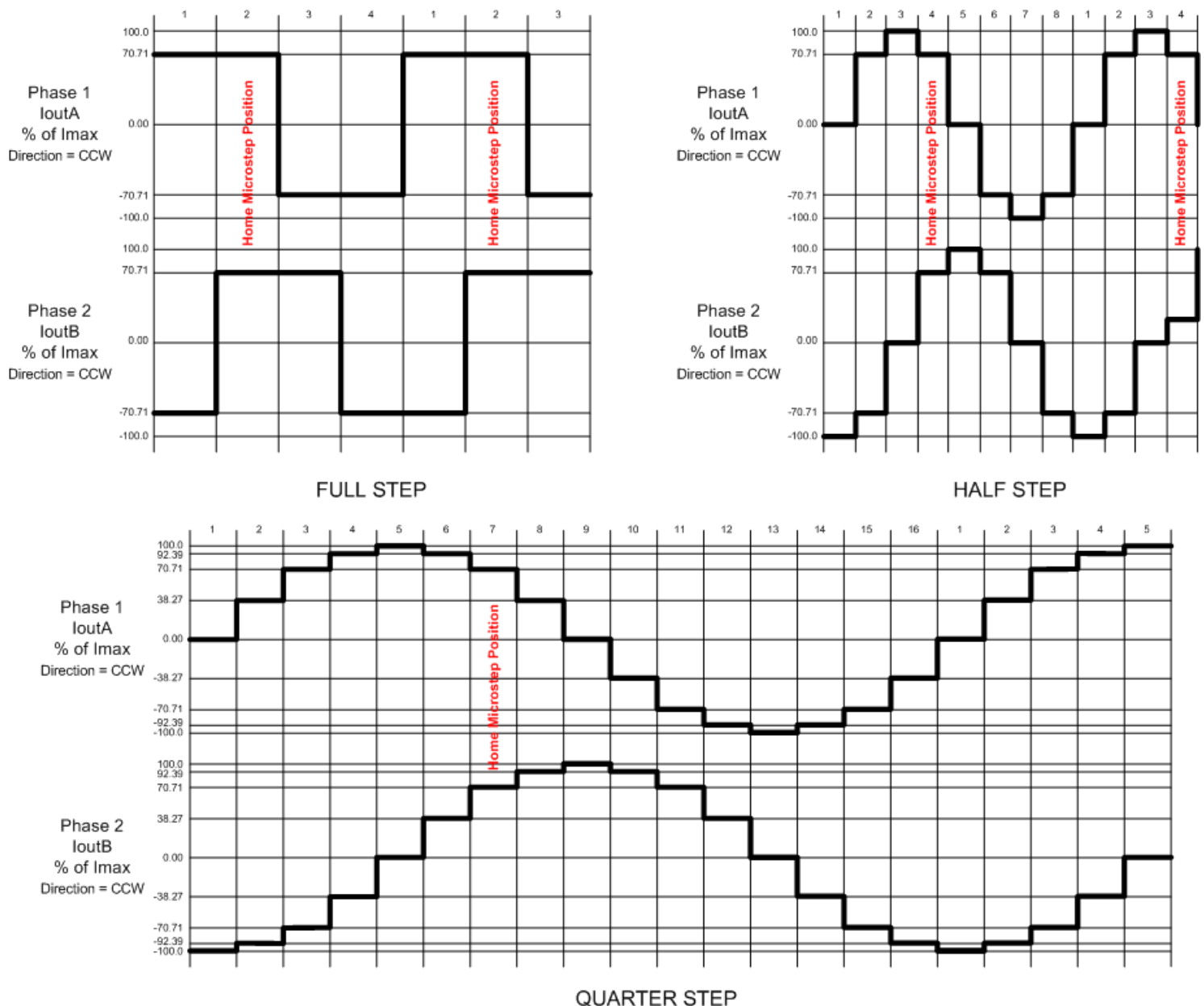
Figure 5 – Full, Half and Quarter-Step Waveforms
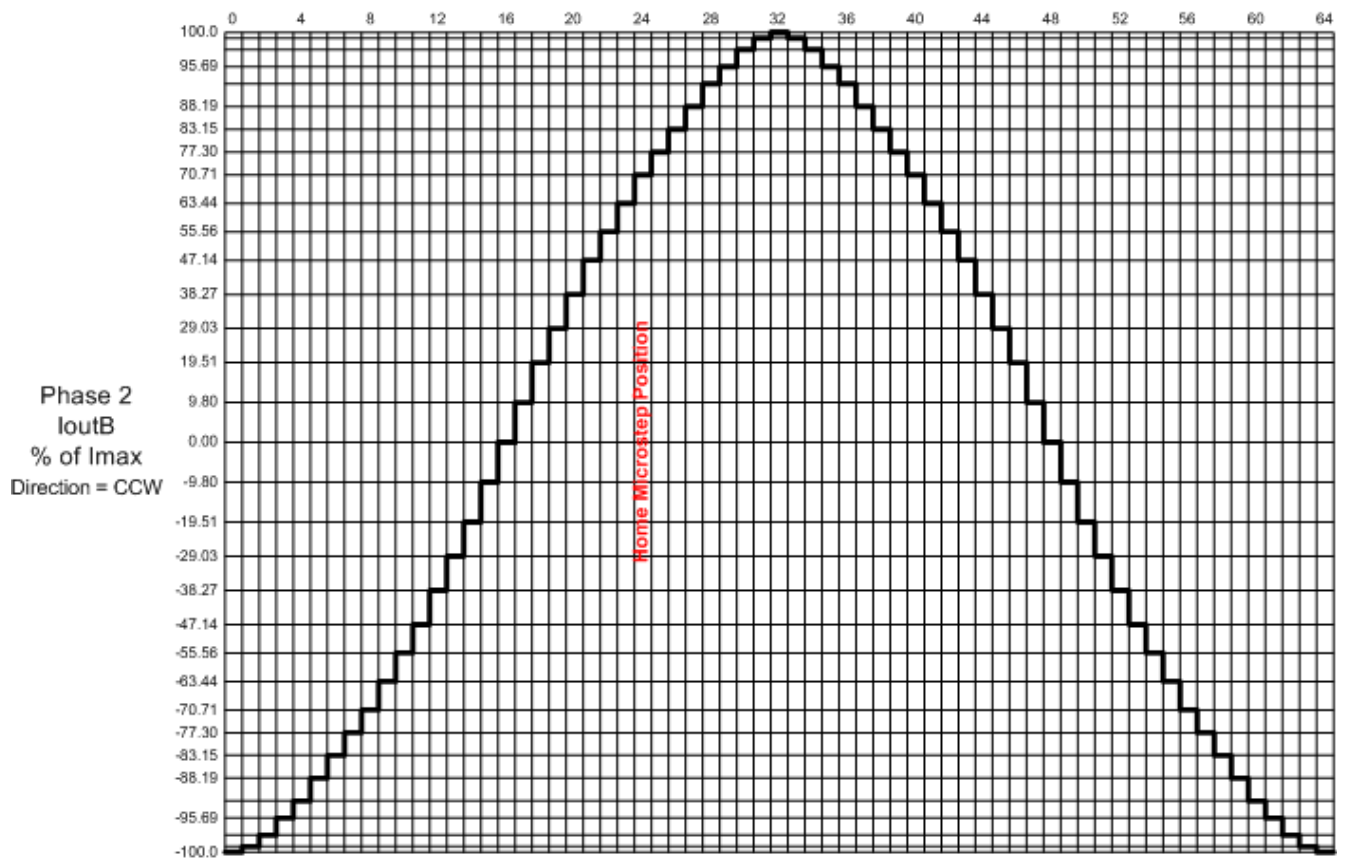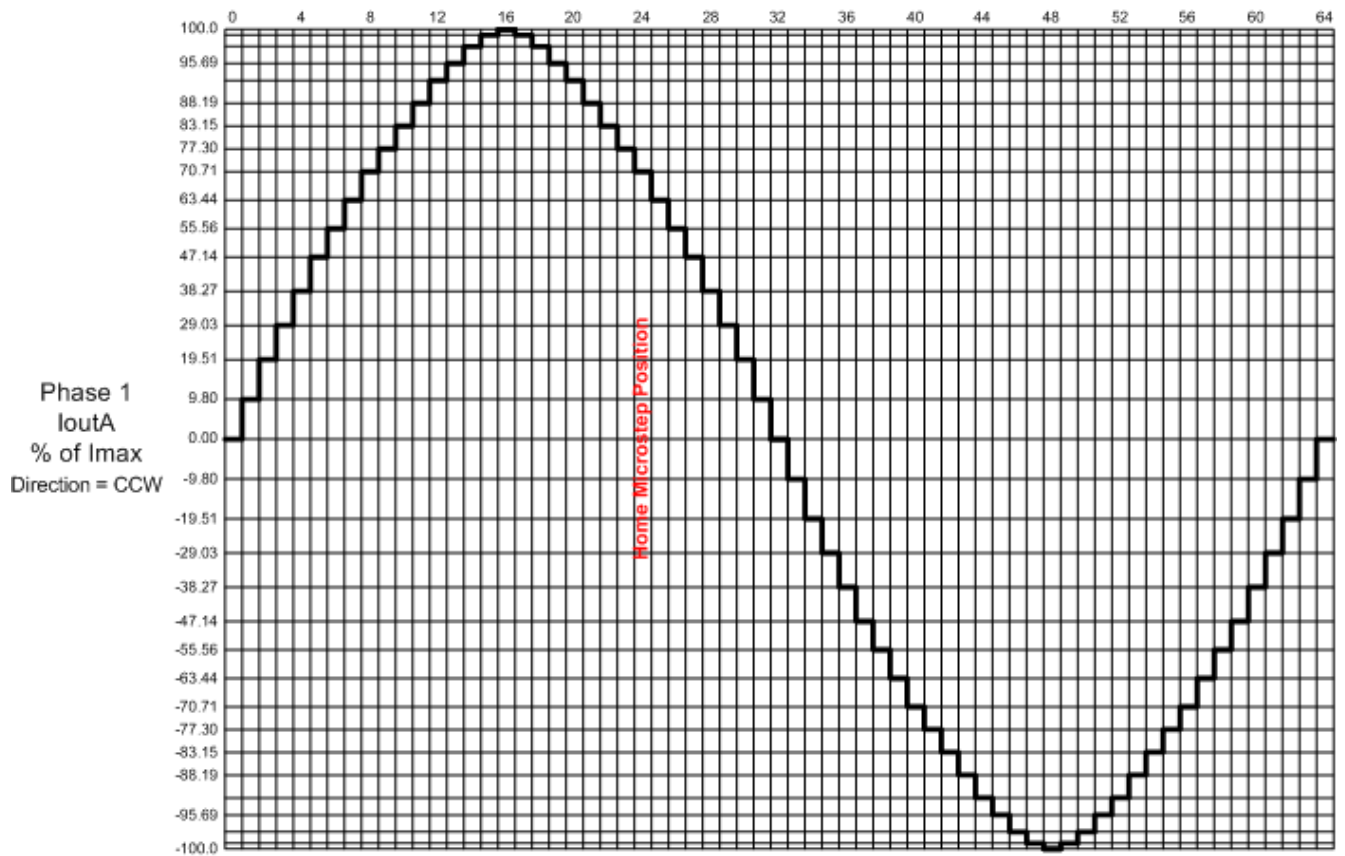
Figure 6 – Sixteenth-Step Waveforms

1.2.9. Percent Fast Decay

This setpoint is used by the driver to control the percent fast decay (PFD), as supported by the driver chip A3979. The follow text was copied directly from the Allegro datasheet explaining how the PFD is used by the chip.

When a STEP input signal commands a lower output current than the previous step, it switches the output current decay to Slow, Fast, or Mixed decay mode, depending on the voltage level at the PFD input. If the voltage at the PFD input is greater than 0.6 ×V_{DD}, then Slow decay mode is selected. If the voltage on the PFD input is less than 0.21 ×V_{DD} , then Fast decay mode is selected. Mixed decay mode is selected when V_{PFD} is between these two levels, as described in the next section. This terminal should be decoupled with a 0.1 µF capacitor.

Mixed Decay Operation. If the voltage on the PFD input is between 0.6 ×V_{DD} and 0.21 ×V_{DD}, the bridge operates in Mixed decay mode, as determined by the step sequence (shown in figures 2 through 5). As the trip point is reached, the device goes into Fast decay mode until the voltage on the RCx terminal decays to the same level as voltage applied to the PFD terminal. The time that the device operates in fast decay is approximated by:

$$t_{FD} = R_T C_T ln (0.6V_{DD}/V_{PFD})$$

After this Fast decay portion, the device switches to Slow decay mode for the remainder of the fixed off-time period.

By default, the PFD used is *'Mixed'*, and the available options for this setpoint, as supported by the driver chip A3979, are listed in the following table.

0	*Slow*
1	*Fast*
2	*Mixed*

Table 5 – Percent Fast Decay Options

1.2.10. Acceleration Table

As mentioned in section 1.2.3, the stepper drive supports acceleration/deceleration lookup tables. If selected, the table is used when the motor is started or stopped, to dynamically change the step/s/s. Up to five different acceleration/deceleration curves are supported.

Acceleration is only used when the motor is being started from a stopped position. Deceleration is only used when the motor is being fully stopped (i.e. to zero) while running. Changes to the step rate during operation, i.e. to make the motor faster or slower, are implemented immediately based on the new rate, without using any acceleration/deceleration.

When accelerating, if the commanded step rate is lower than the maximum step rate at the end of the acceleration table, then the motor will only accelerate to the commanded rate and then continue to spin at that speed.

For example, using the example table shown in Figure 7, if the commanded step rate is 48 Steps/s, and the motor is started, then the acceleration table will be followed for the first 50ms. It will then stop at 48 Steps/s 60ms after the '*Running'* motor mode command was received.

Time [Sec]	Velocity [Step/s]
0.000	0
0.010	15
0.020	25
0.030	33
0.040	40
0.050	46
0.060	51
0.070	55
0.080	58
0.090	60
0.100	62
0.110	63
0.120	64
0.130	65
0.140	65
0.150	65
0.160	65
0.170	65
0.180	65
0.190	65
0.200	65
0.210	65
0.220	65
0.230	65
0.240	65
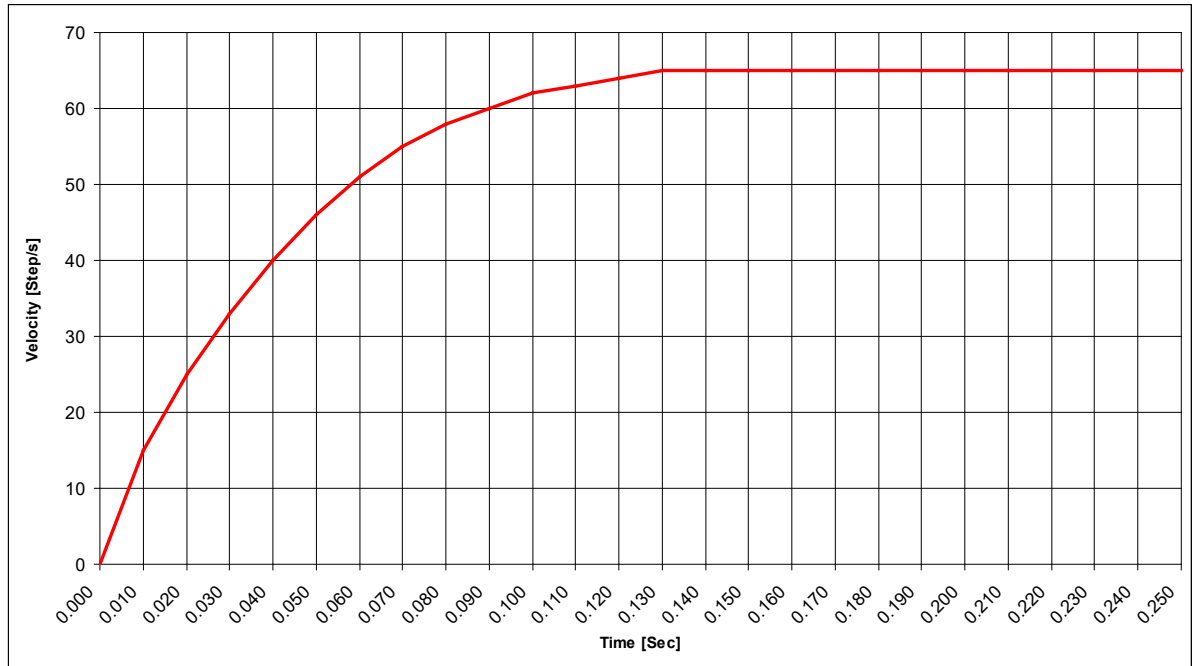
Figure 7 – Example Acceleration Curve

Conversely, if the motor was running at 48 Steps/s when the '*Stopped'* motor mode command was received, then the drive would establish where the motor was running on the deceleration curve, and decelerate to zero from there. As with the starting operation, stopping would also take 60ms in the case of the example.

However, should the motor have been running at 70 Steps/s when the 'Stopped' command was received, it would immediately jump to the maximum rate on the curve, in this case 65 Steps/s and hold there for 120ms (130ms to 250ms). It would then decelerate to zero, changing the rate every 10ms as shown on the curve. To avoid the delay at the beginning, the last 12 entries on the table could have all been set to 130ms in the Time column.

To select an acceleration/deceleration profile, a non-zero **"Acceleration Table"** must be commanded. Otherwise, the drive will simply jump to or from the commanded **"Step Rate"** as the **"Motor Mode"** changes from '*Stopped'* to '*Running'* or vice versa. By default, no acceleration table is used, and the available options for this setpoint, as supported are listed in the following table.

0	*NoProfile*
1	*Profile1*
2	*Profile2*
3	*Profile3*
4	*Profile4*
5	*Profile5*

Table 6 – Acceleration Table Options

1.2.11. Put Drive to Sleep

A feature of driver chip A3979 is that it can be put into sleep mode to draw a low current when it is not operating. To utilize this feature, the drive can receive a command from the J1939 network to put the drive to sleep.

This input is treated as a Boolean value by the drive, so the data received is interpreted as either "False" (default, drive is awake) or "True" (drive is put to sleep). By default, the corresponding CAN Receive message has its default data set with **"Received Data Min (Off Threshold)"** as 0.0 and **"Received Data Max (On Threshold)"** as 1.0.

While the **"Put Drive to Sleep"** command is received as 1 (or greater), the drive will be disabled, and the chip is in low-power sleep mode. To reset the drive after the command has cleared (gone back to zero, or lower), the **"Motor Mode"** must transition from *'Stopped'* to *'Running'*. Alternatively, one of the other command data messages, such as the step rate, must change.

1.2.12. Reset Motor Drive

A feature of driver chip A3979 is that it reset the output current to the microstep "Home" position as shown in Figures 5 and 6. To utilize this feature, the drive can receive a command from the J1939 network to reset the drive to "Home." While in reset mode, the motor will not be running.

This input is treated as a Boolean value by the drive, so the data received is interpreted as either "False" (default, is not reset) or "True" (drive is reset to home). By default, the corresponding CAN Receive message has its default data set with **"Received Data Min (Off Threshold)"** as 0.0 and **"Received Data Max (On Threshold)"** as 1.0.

While the **"Reset Motor Drive"** command is received as 1 (or greater), the drive will be disabled, and the chip will begin micro-stepping from the home position when it restarts. To re-enable the drive after the command has cleared (gone back to zero, or lower), the **"Motor Mode"** must transition from *'Stopped'* or *'Holding'* to *'Running'*. Alternatively, one of the other command data messages, such as the step rate, must change.

1.2.13. E-Stop Motor Drive

Another way to fully disable the motor drive (i.e. ensure that there is zero current running through both phases) is to use the Emergency Stop (E-Stop) input. By default, this input is mapped to a CAN Receive message, but it could also be linked to one of the on-board inputs configured as a digital input.

When controlled by a CAN message, this input is treated as a Boolean value by the drive, so the data received is interpreted as either "False" (default, E-Stop is OFF) or "True" (E-Stop ON, drive disabled). By default, the corresponding CAN Receive message has its default data set with **"Received Data Min (Off Threshold)"** as 0.0 and **"Received Data Max (On Threshold)"** as 1.0.

This signal is a true override signal, such that while **"E-Stop Motor Drive"** is active (ON), the drive will not run. To re-enable the drive after the E-Stop has cleared (gone back to zero, or lower), the **"Motor Mode"** must transition from *'Stopped'* to *'Running'*.

1.2.14. Save Data to Defaults

It may be desirable, in situations where the Axiomatic EA is not connected to the network, to be able to change the power-up default values used by the stepper drive by simply sending a CAN message. In order to support this feature, there is an option to save the current motor drive settings to non-volatile memory by simply setting a CAN bit.

This input is treated as a Boolean value by the drive, so the data received is interpreted as either "False" (no command to save) or "True" (save data). By default, the corresponding CAN Receive message has its default data set with **"Received Data Min (Off Threshold)"** as 0.0 and **"Received Data Max (On Threshold)"** as 1.0.

The save operation will only occur when the **"Save Data to Defaults"** bit transitions from a low (OFF) to a high (ON).

1.3. PID Control Function Block

The PID Control Function block can be used to control the **"Motor Mode"**, **"Direction"** and **"Step Rate"** commands as described in Section 1.2.3.
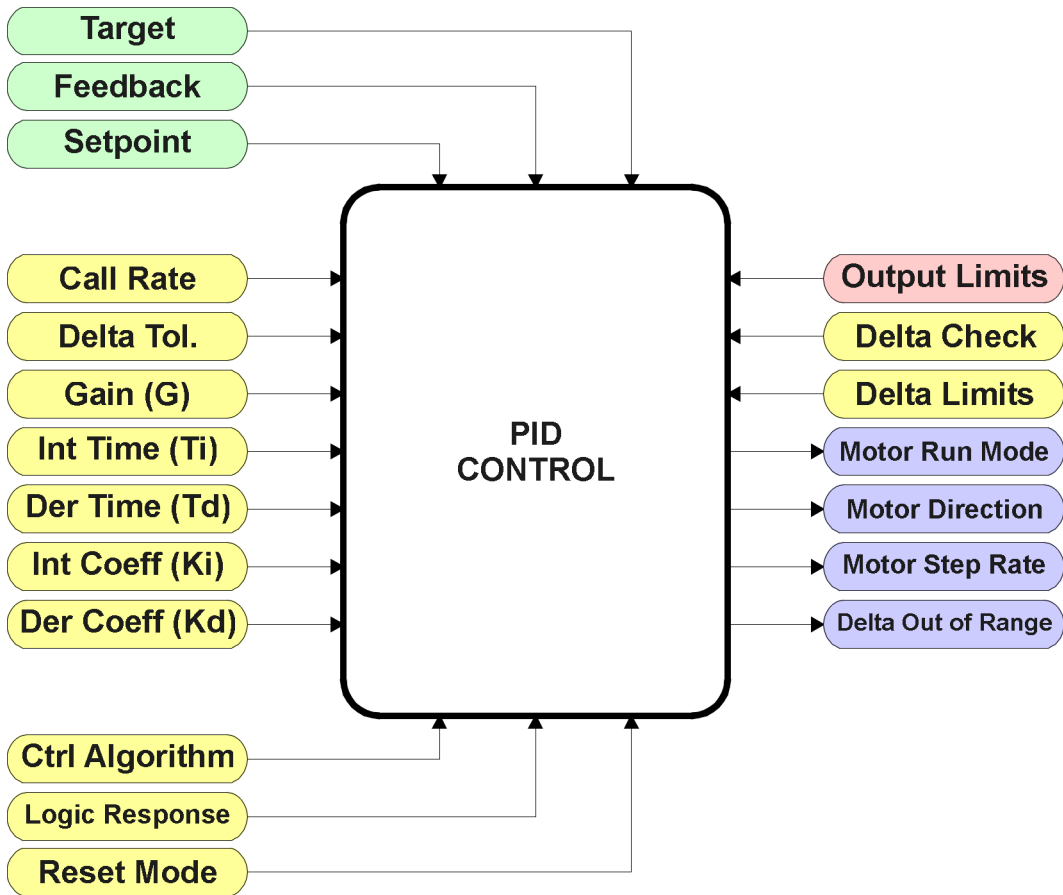
Figure 8 – PID Control Function Block

When used, the **"Motor Mode"** and **"Direction"** will automatically be controlled by the PID function, depending on the **"PID Logic Response"** selected. The PID output will be some calculated value between 0 to 100% which will be used as the X-axis input to the **"Step Rate"** control (see Section 1.2.5.)

Target versus Setpoint Control

There are two types of **"PID Control Algorithm"** that can be selected to establish how the PID function will response to the Feedback input, as listed in the following table.

0	Target Control
1	Setpoint Control

Table 7 – PID Control Algorithm Options

When the *'Target Control'* option is used, the PID loop will use the value of the Target input, as set by the **"PID Target Command Source"** and **"Number"** setpoints to compare to the Feedback input, as set by the **"PID Feedback Command Source"** and **"Number"** setpoints.

Both inputs will be normalized to 0 to 100%, using the input limits to set the range. For example, if the Target input is selected using a CAN Receive message, **"Received Data Min"** corresponds to 0% while **"Received Data Max"** is treated as 100% in the linear calculation defined in Figure 4. However, should the Target be controlled using the Frequency/PWM Input 1, then the input's **"Minimum Range"** corresponds to 0%, while the **"Maximum Range"** is 100%. Similarly, the same is true for the Feedback input. Because of the normalization to a percentage of the signal, it is not required that the Target and Feedback inputs be in the same units (i.e. both voltage.)

However, if the *'Setpoint Control'* option is used, it is assumed that the **"PID Setpoint Control Target"** value will be defined in the same unit as the Feedback input. It too will be normalized to a percentage, using the limits of the Feedback input to determine the range for the 0 to 100% linearization calculation. (Note: When setpoint control is used, the command source setpoints are not used in the algorithm, although they still show up as available in the Axiomatic EA.)

In order to allow the output to stabilize, the user can select a non-zero value for "**PID Delta Tolerance**". If the absolute value of Error$_K$ is less than this value, Error$_K$ in the formula shown in Figure 9 will be set to zero. The units for the tolerance are also assumed to be the same as the Feedback input. Like the setpoint value, the tolerance will be normalized to a percentage using the Feedback input range.

PID Control Algorithm

As described above, all the values used in the PID Control Algorithm are handled as a percentage value when inputted into the formula.

The PID algorithm used is shown below, with names in red being the configurable parameters.

$$T = Loop\_Update\_Rate*0.001$$

$$P\_Gain = G$$
$$I\_Gain = G*Ki*T/Ti$$
$$D\_Gain = G*Kd*Td/T$$
$$Note: If\ Ti\ is\ zero,\ I\_Gain = 0$$

$$Error_k \quad = Target - Feedback$$
$$ErrorSum_k = ErrorSum_{k-1} + Error_k$$

$$P_k = Error_k * P\_Gain$$

$$I_k = ErrorSum_k * I\_Gain$$

$$D_k = (Error_k - Error_{k-1}) * D\_Gain$$

$$PIDOutput_k = P_k + I_k + D_k$$

Figure 9 – PID Control Algorithm

When active, the PID algorithm will be called every "**PID Loop Update Rate**" (in ms)

Each system will have to be tuned for the optimum motor response. Response times, overshoots and other variables will have to be decided by the customer using an appropriate PID tuning strategy.

PID Control Logic

It is recognized that in some systems, it could be desirable to have a push-pull type response. That is, the motor will run in one direction (pushed) when under target, and runs in the other direction when over (pulled). In other systems, it may be that the motor is expected to run at some particular speed to maintain a target variable. For this reason, there is a setpoint **"PID Logic Response"** which will define how the motor will respond to changes in the Target or Feedback inputs. The options for this setpoint are listed in the following table.

0	*Run Motor Unidirectional*
1	*Motor CW when Under Target*
2	*Motor CW when Over Target*

Table 8 – PID Logic Response Options

When running in the first mode *'Run Motor Unidirectional'*, a traditional PID response will occur, where the motor will run faster the farther away the feedback is from the target, and will slow down as the difference decreases. Once the feedback signal is within tolerance of the target/setpoint, the PID output will remain constant at whatever value it was at when the signals converged.

In this type of control, the direction of the motor will always be as per defined in the **"Default Direction."** The motor mode will always be '*Running'* (can still be stopped using the E-Stop input).

However, in a push-pull type system (selections 1 or 2 from table 8); a purely proportional response at the output is likely more desirable than a PI or PID response. As the feedback signal goes farther away from the target, the motor should run faster to bring it back in alignment sooner. As the signals start to converge, however, the motor should slow down so as to not overshoot the target. For this reason, the **"PID Integral Coefficient, Ki"** and the **"PID Derivative Coefficient, Kd"** should be set to zero to get a P-only response. The **"PID Gain Coefficient, G"** can be adjusted to change the error versus speed response of the system.

In this type of control, the direction of the motor will be determined by the delta error (difference) between the target/setpoint value and the feedback signal. Options 1 and 2 in table 8 are identical in everyway expect for which direction the motor will rotate when the error is positive or negative.

Reset Response

Typically, the motor will be *'Running'* while the PID algorithm is being used. However, as explained earlier, when the error between the target and feedback values are within the "**PID Delta Tolerance**" value, the PID function is reset, and the Step Rate for the motor is set to 0.

In this instance, the **"PID Reset Response"** determines if the motor will be '*Disabled'*, '*Stopped'* or '*Holding'* while the PID loop is idle. The default response is '*Holding*.'

<u>Delta Checking</u>

In some instances, it may be desired to have an indicator if the delta error, determined by the difference between the (*Target – Feedback*) inputs, goes beyond some threshold. When the Delta value goes out of range, a flag will be set (0 or 1), and this flag can be used as an input to another function block, i.e. the Digital Output, or a CAN Transmit message.

By default, this feature is disabled, however Table 9 outlines the options that can be selected for the **"PID Delta Check Response"** to activate the indicator.

0	*Delta Checking Disabled*
1	*Set when (Target-Feedback) more than limit*
2	*Set when (Target-Feedback) less than limit*
3	*Set when Absolute (Target-Feedback) more than limit*
4	*Set when Absolute (Target-Feedback) less than limit*

Table 9 – PID Delta Check Response Options

The setpoints **"PID Delta Check Set Limit"** and **"PID Delta Check Clear Limit"** are used only when the response is set to a non-zero value. Like the **"PID Delta Tolerance"**, the units for these setpoints are also assumed to be the same as the Feedback input and they will be normalized to a percentage using the Feedback input range.

The two limit values are provided in order to provide some hysteresis for the *Delta Out Of Range* flag. If option 1 or 3 is selected, **"PID Delta Check Set Limit"** must be higher than **"PID Delta Check Clear Limit."** When option 2 or 4 is used**,** the **Clear** limit must be higher than the **Set** limit. By default, both limits are set to zero.

<u>Default Configuration</u>

By default, the PID Control Function is not used. However, should it be selected, the default settings are that it will use the voltage input as the feedback signal, and be setup for a *'Setpoint Control'* response, targeting 2.5V. There will be a +/- 100mV allowable tolerance between setpoint and feedback. The motor will rotate clockwise when the feedback value is less than the setpoint, and counter-clockwise when the feedback is too large. The output will respond proportionally by the gain value, G, as the feedback and target values diverge or converge. The Ki and Kd setpoints are set to 0.0 to disable the integral and derivative portions of the response.

See Section 4.3 for more information about the default settings.

1.4. Input Function Blocks

There are two types of input function blocks, one for frequency/PWM and the other for voltage/current inputs. Both input function blocks are virtually identical whether measuring an ADC or Timer input, as shown below.
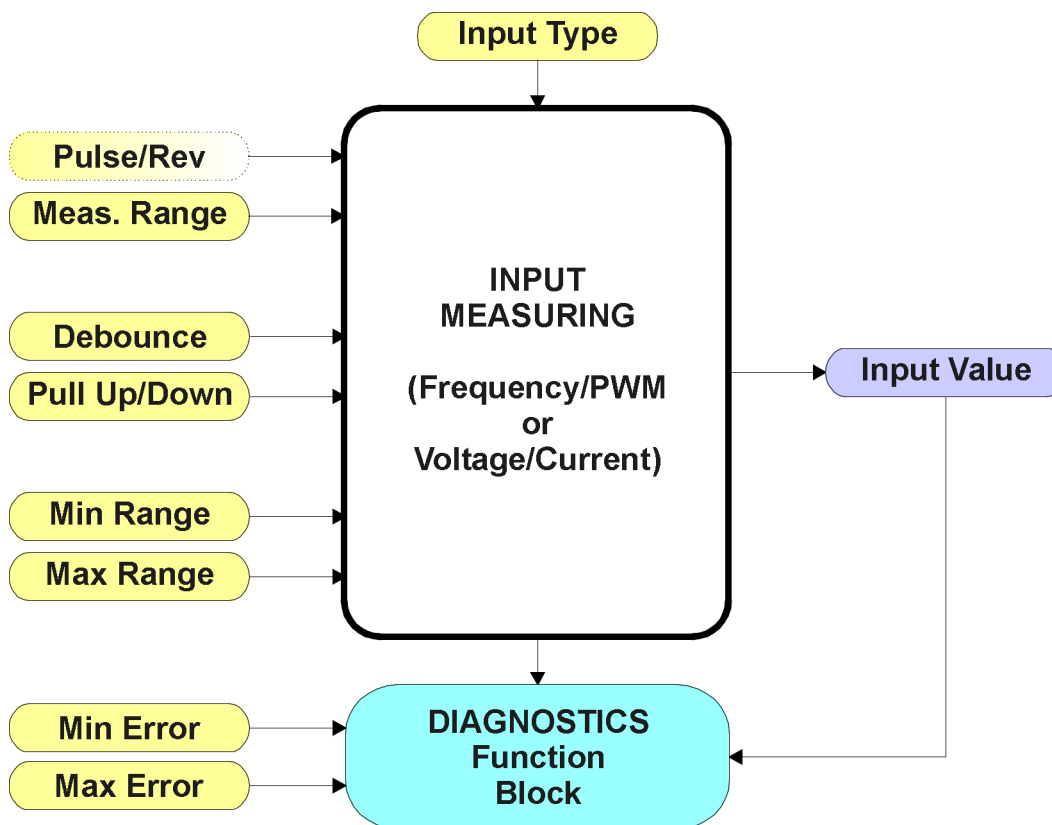
Figure 10 – Input Function Block

The **"Input Type"** is the most important setpoint associated with this function block, and it should be selected first. Changing it will result in other setpoints being automatically updated by the drive to match the new type.

Input Circuits

The Analog Input can be selected as either a Voltage or Current input. It has a hardware filter that eliminates the effects of noise on the input, which then goes directly to the pin of a 12-bit analog-to-digital converter (ADC.) A voltage input is a high impedance (1MΩ) input protected against shorts to GND or Vcc. In current mode, a 130Ω resistor is used to measure the input signal. The Analog Input should be connected to the AGND reference pin provided on the connector.

The following **"Analog Input Type"** options can be selected for Voltage/Current Input 1.

0	Disabled
1	Voltage
2	Current
3	Digital

Table 10 – Analog Input Type Options

The Frequency Input can read either a Frequency/RPM or Pulse Width Modulated (PWM) input. The circuit is designed to read either an AC input signal, such as one from an inductive pickup sensor, or an active signal swinging between +Vcc and GND, such as from a Hall Effect sensor. For active inputs, a 10kΩ pull-up or pull-down resistor could also be connected at the input to handle open-collector NPN (active low, pull-up) or PNP (active high, pull-down) signals. The conditioned input signal is then connected to a 16-bit timer pin on the processor. It is fully independent of the Analog Input.

The following **"Frequency Input Type"** options can be selected for Frequency/PWM Input 1.

0	*Input Disabled*
1	*Inductive Pickup*
2	*Active Freq/RPM*
3	*PWM Duty Cycle*
4	*Digital Signal*

Table 11 – Frequency Input Type Options

As can be seen from tables 10 and 11, both input types could also be configured as a Digital Input. In either case, an active high or low signal could be read by selecting the appropriate pull-up or pull-down 10kΩ resistor, as shown in the following table.

0	*Pullup/down Off*
1	*10kOhm Pullup*
2	*10kOhm Pulldown*

Table 12 – Pullup/Pulldown Resistor Options

A regulated +5V is provided to power an input such as a potentiometer. It is referenced with respect to the AGND used by the Analog Input, and is capable of sourcing up to 10mA.

Measuring Data

Each input type has associated with it a configurable **"Input Measuring Range."** This setpoint allows the user to optimize the hardware for the input signal.

For voltage inputs, there are actually four input ranges that can be selected. Typically, inputs will be in either the 0-5V or 0-10V range, but for more resolution at smaller voltages, a low range can be selected when appropriate. However, this hardware does not actually support the 0-1V input range, so if it selected, the drive will default back to a 0-5V range.

0	*0 to 1V*
1	*0 to 2.5V*
2	*0 to 5V*
3	*0 to 10V*

Table 13 – Voltage Input Ranges

For current inputs, both ranges are the same at the hardware level. However, in 4-20mA, the minimum value cannot be set below 4mA.

0	0 to 20mA
1	4 to 20mA

Table 14 – Current Input Ranges

For frequency inputs, the user can select the following ranges shown in the following table. Beside it, the overflow values are shown, showing how long it would take to flag an error once the signal is lost (i.e. no frequency measured.)

		Overflow [Sec]
0	0.5Hz to 50Hz	~4 [0.25Hz]
1	10Hz to 1kHz	~0.105 [9.5Hz]
2	100Hz to 10kHz	~0.010 [95Hz]

Table 15 – Frequency Input Ranges

The means by which the frequency is calculated is by measuring the time between pulses. The advantage of this type of measuring technique is that it gives an accurate reading of low frequency signals, with 2 decimal places of resolution. However, it has a drawback in that it will take several seconds to detect a loss of signal when the pulses go away, and may be read as a very high frequency signal during the transition phase. Therefore, it is highly recommended that diagnostics on low frequency inputs are not sent to the network for at least 5 seconds, to ensure that a low signal error is flagged, not a high error. If a signal will never go below 10Hz, it is highly recommended to use a more appropriate range for faster error reaction.

For PWM Inputs, there are only two types of measuring ranges, as shown below. To guarantee the rated accuracy from 1% to 99%, ensure that the appropriate measuring range is selected.

0	Low Freq (<1kHz)
1	High Freq (>100Hz)

Table 16 – PWM Input Ranges

The **"Minimum Range"** and **"Maximum Range"** setpoints must not be confused with the measuring range. These setpoints are available with all but the digital input, and they are used when the input is selected as a control input for another function block. They become the Xmin and Xmax values used in the slope calculations (see Figure 4). When these values are changed, other function blocks using the input as a control source are automatically updated to reflect the new X-axis values.

Diagnostic Detection

The **"Minimum Error"** and **"Maximum Error"** setpoints are used with the Diagnostic function block. The values for these setpoints are constrained such that

  0 <= Minimum Error <= Minimum Range <= Maximum Range <= Maximum Error <= 1.1xMax*

* The maximum value for any input is dependent on type. The error range can be set up to 10% above this value.

    Frequency: Max = 10,000 [Hz or RPM]
    PWM: Max = 100.00 [%]
    Voltage: Max = 1.00, 2.50, 5.00 or 10.00 [V]
    Current: Max = 20.00 [mA]

More information about the Diagnostic function block can be found in Section 1.7.

Digital Inputs

Since either input could be configured to act as a digital input, they both have this as an option for **"Input Type."** In this setup, when using it as an input to another function block (i.e. an E-Stop signal), map the control to the *'Digital Input Read State'* type instead of the *'Input Measured Value.'* Here, the analog input becomes Digital Input 1 and the Frequency Input becomes Digital Input 2.

In contrast to the analog input types, the digital input function is very simple, as shown below.
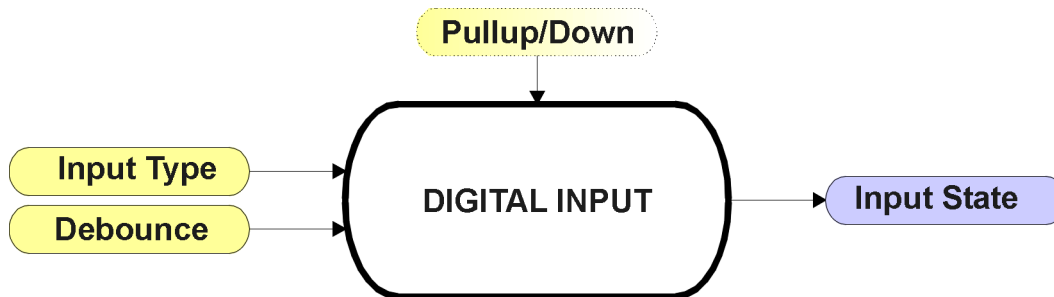
Figure 11 – Digital Input Function Block

To prevent spurious signals from affecting the logic of the controller, all digital inputs have associated with them a debounce time. If and only if the signal at the end of the debounce period has changed state will the change of state of the input be translated to the controller. The **"Debounce Time"** setpoint can be selected from the table below.
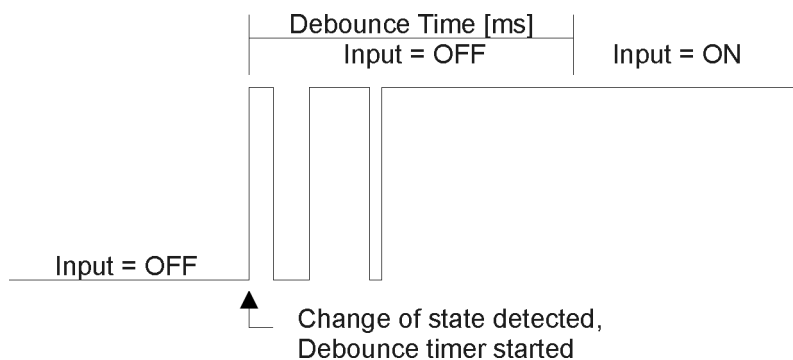
Figure 12 – Digital Input Debouncing

0	*None*
1	*111ns*
2	*1.78us*
3	*14.22us*

Table 17– Debounce Time Options

In this case, the **"Input Measuring Range"** setpoint determines the Input State used by other function blocks based on the actual state of the input at the pin, per the table below.

		Active High	**Active Low**	**State**
0	*Normal On/Off*	HIGH	LOW	ON
		LOW or Open	HIGH or Open	OFF
1	*Inverse Logic*	HIGH	LOW	OFF
		LOW or Open	HIGH or Open	ON
2	*Latched Logic*	HIGH to LOW	LOW to HIGH	No Change
		LOW to HIGH	HIGH to LOW	State Change (i.e. OFF to ON)

Table 18 – Digital Input Type versus Input State

1.5. Digital Output Function Block

There is a single output on the stepper drive, capable of sourcing up to 5A that can be used to drive a digital load such as a relay or small lamp.
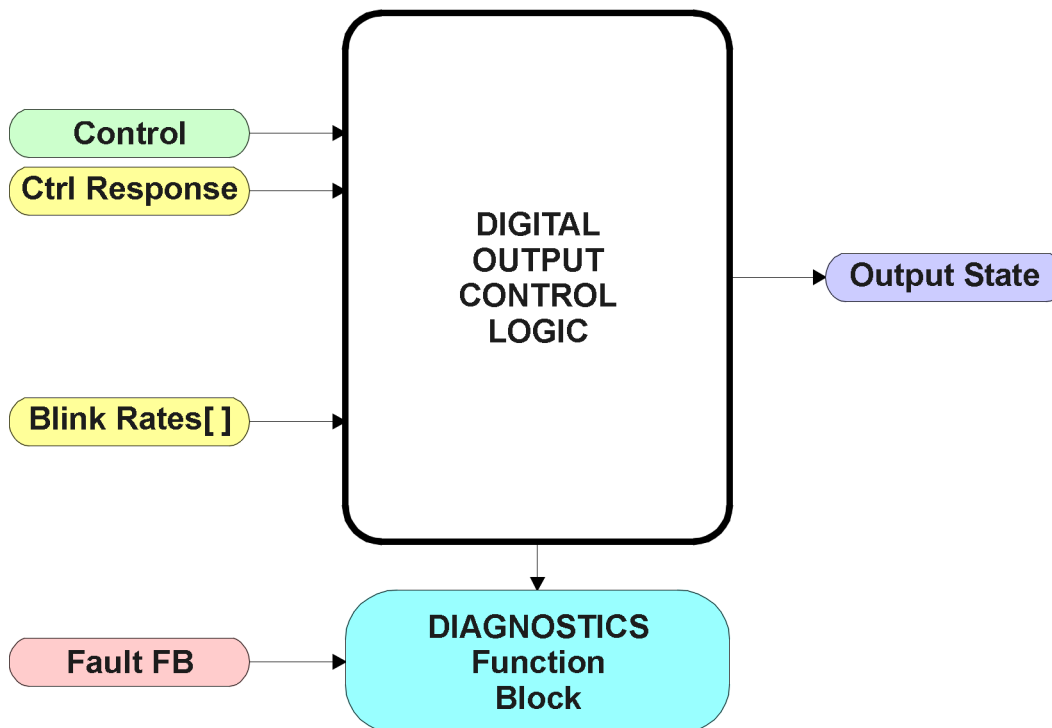
Figure 13 – Digital Output Function Block

Since this is a digital output (Dout), it is assumed that the **"Control Source"** will be a digital input, and by default it is setup to respond to a CAN Receive message.

However, if a non-digital control is selected, the command state will be OFF at or below the minimum input, ON at or above the maximum input, and it will not change in between those points. In other words, the input will have built in hysteresis, as shown in Figure 14. ***This relationship is true for any function block that has a non-digital input mapped to a digital control.***
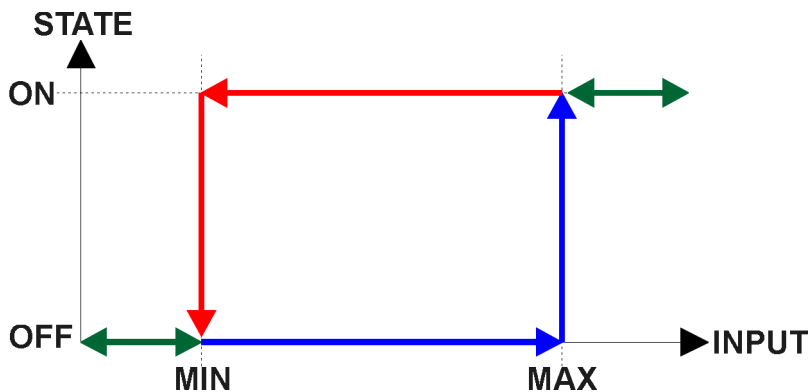
Figure 14 – Analog to Digital Input

There are several different ways that the output will respond to the data from the command source, as outlined in the following table.

0	*Disabled*
1	*Blinking*
2	*Normal*
3	*Inverse*
4	*Latched*

Table 19 – Digital Control Responses

The Digital Output function block is unique in that the **"Control Source"** here can be selected to *'Reflect When a Fault is Detected'*. This option allows the digital output to be setup as an Error Lamp, which will blink at different rates should a fault be flagged. It has associated with it three blink rates for this mode to reflect **"Motor"**, **"Input"** or **"Other"** faults.

Otherwise, the **"Control Response"** setpoint will determine the actual output state as compared to the commanded state.

In a *'Normal'* response, when the control input commands the output ON, then the output will be turned ON. However, in an *'Inverse'* response, the output will be ON unless the input commands the output ON, in which case it turns OFF.

If a *'Blinking'* response is selected, then while the input commands the output ON, it will blink at the rate in the **"Motor Fault Blink Rate"** setpoint. When commanded OFF, the output will stay off.

If a *'Latched'* response is selected, only when the input commands the state from OFF to ON, will the output change state.

1.6. Filtering Function Block

The filtering function block is associated with phase current feedback. When used, the software filter is applied to the measured current feedback value for each of the drive phases before it is used for error detection and as a feedback signal input to other function blocks (i.e. CAN transmission)
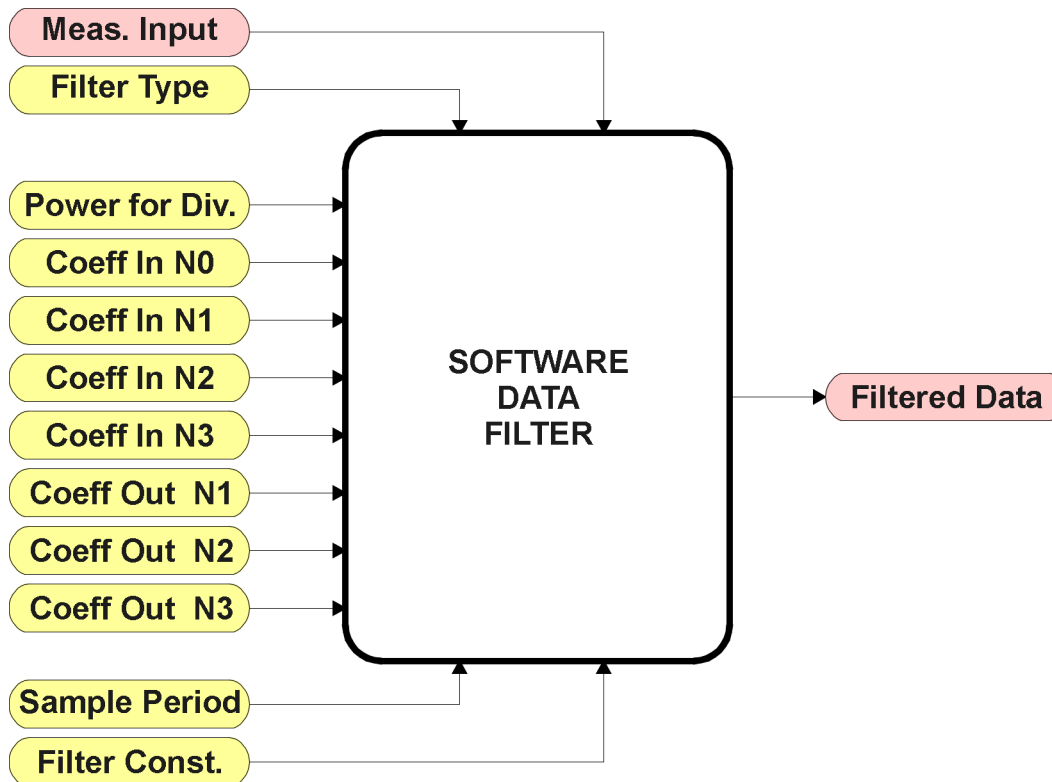
Figure 15 – Software Data Filter Function Block

The **"Software Filter Type"** is the most important setpoint associated with this function block, and it should be selected first. Changing it will result in other setpoints being enabled/disabled as appropriate for the selected filter. The available filters types are shown in the following table.

0	*No Filter*
1	*Moving Average*
2	*Repeating Average*
3	*3rd Order Low Pass*

Table 20 – Software Filter Types

By default, the Phase Current Filter is setup as a *'Repeating Average'* type with a **"Software Filter Constant"** of 100. The output from the filter block does NOT represent the true RMS value of the phase currents. Rather, it is used solely for error detection to flag when a phase is open-circuited.

The other filter options are described on the next page.

Calculation with no filter:

Value = Input

The data is simply a 'snapshot' of the latest value measured by the ADC or timer.

Calculation with the moving average filter:

$$Value_N = Value_{N-1} + \frac{(Input - Value_{N-1})}{FilterConstant}$$

This filter is called every 1ms.

Calculation with the repatng average filter:

$$Value = \frac{\sum Input_N}{N}$$

At every reading of the input value, it is added to the sum. At every N^{th} read, the sum is divided by N, and the result is the new input value. The value and counter will be set to zero for the next read. The value of N is stored in the "**Filter Constant**" setpoint. This filter is called every 1ms.

Calculation with the 3rd Order Low Pass filter:

$$Value_N = \frac{[C_{IN0}*Input_N + C_{IN1}*Input_{N-1} + C_{IN2}*Input_{N-2} + C_{IN3}*Input_{N-3} + C_{OUT1}*Value_{N-1} + C_{OUT2}*Value_{N-2} + C_{OUT3}*Value_{N-3}]}{2^{Power}}$$

C_{IN0} Coefficient for Input (n)
C_{IN1} Coefficient for Input (n-1) C_{OUT1} Coefficient for Output (n-1)
C_{IN2} Coefficient for Input (n-2) C_{OUT2} Coefficient for Output (n-2)
C_{IN3} Coefficient for Input (n-3) C_{OUT3} Coefficient for Output (n-3)

This filter uses 16-bit fixed point math. The "**Filter Power for Divisor**" setpoint tells the controller the shift value used when the coefficients were selected.

The filter will be called every "**Filter Sampling Period**" (in ms). In between calls, the data of the input/feedback signal is the value which was calculated the last time the filter was called.

1.7. Diagnostic Function Block

There are several types of diagnostics supported by the Stepper Drive. Fault detection and reaction is associated with either input as well as the motor and output drives. In addition to I/O faults, the drive can also detect/react to power supply and lost communication events.
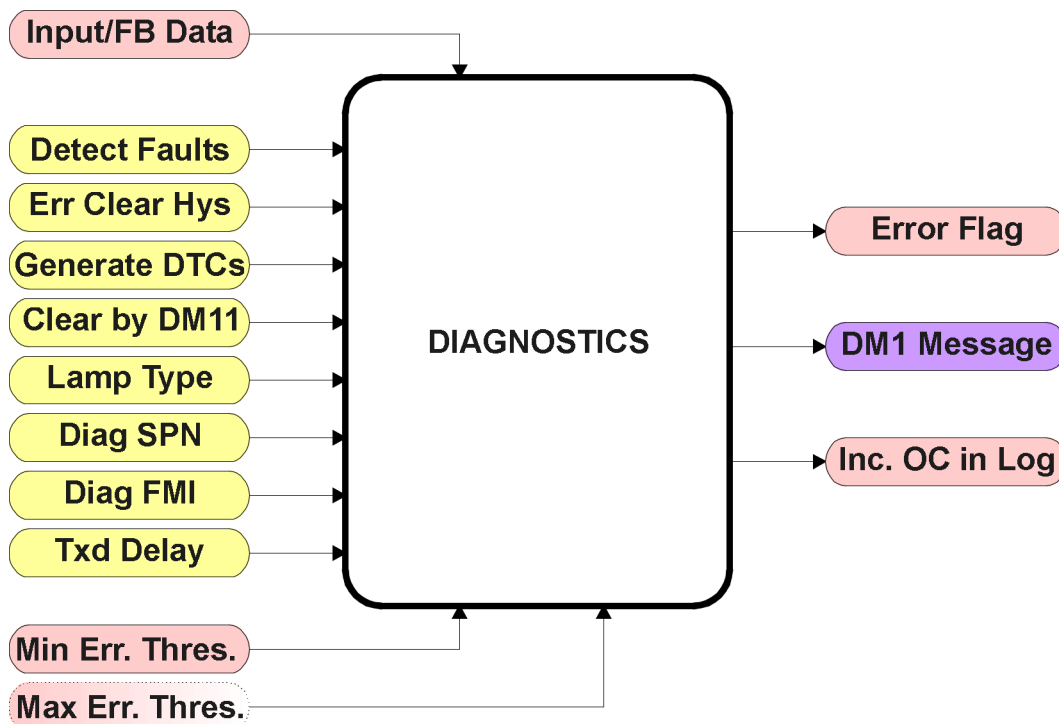
Figure 16 – Diagnostics Function Block

The **"Fault Detection is Enabled"** is the most important setpoint associated with this function block, and it should be selected first. Changing it will result in other setpoints being enabled/disabled as appropriate. When disabled, all diagnostic behaviour associated with the I/O or event in question is ignored (i.e. this type of fault will not disable the motor)

In the case of a power supply error, it can be selected to disable the stepper motor when detected. By default, a power supply under or over voltage condition will shutoff the motor.

In most cases, faults can be flagged as either a low or high occurrence. The min/max thresholds for all diagnostics supported by the stepper drive are listed in Table 21. Bolded values are user configurable setpoints. Some diagnostics react only to a single condition, in which case a N/A is listed in one of the columns.

Function Block	Minimum Threshold	Maximum Threshold
Phase X Winding	**Current Open Circuit Threshold**	N/A
[Continuous] Input	**Minimum Error**	**Maximum Error**
Digital Output	N/A	Fault Feedback Signal (Hardware)
Power Supply	**Power Undervoltage Threshold**	**Power Overvoltage Threshold**
Lost Communication	N/A	**Received Message Timeout** (any)
Hardware Shutdown	N/A	~2.5A (Hardware, Drive Chip)

Table 21 – Fault Detect Thresholds

When applicable, a hysteresis setpoint is provided to prevent the rapid setting and clearing of the error flag when an input or feedback value is right near the fault detection threshold. For the low end, once a fault has been flagged, it will not be cleared until the measured value is greater than or equal to the Minimum Threshold **+ "Hysteresis to Clear Fault."** For the high end, it will not be cleared until the measured value is less than or equal to the Maximum Threshold **- "Hysteresis to Clear Fault."** The minimum, maximum and hysteresis values are always measured in the units of the fault in question.

The next setpoint in this function block is the **"Event Generates a DTC in DM1."** If and only if this is set to true will the other setpoints in the function block be enabled. They are all related to the data that is sent to the J1939 network as part of the DM1 message, Active Diagnostic Trouble Codes.

A Diagnostic Trouble Code (DTC) is defined by the J1939 standard as a four byte value which is a combination of:

SPN	Suspect Parameter Number	(first 19 bits of the DTC, LSB first)
FMI	Failure Mode Identifier	(next 5 bits of the DTC)
CM	Conversion Method	(1 bit, always set to 0)
OC	Occurrence Count	(7 bits, number of times the fault has happened)

In addition to supporting the DM1 message, the Stepper Drive also supports

DM2	Previously Active Diagnostic Trouble Codes	**Sent only on request**
DM3	Diagnostic Data Clear/Reset of Previously Active DTCs	**Done only on request**
DM11	Diagnostic Data Clear/Reset for Active DTCs	**Done only on request**

So long as even one Diagnostic function block has **"Event Generates a DTC in DM1"** set to True, the Stepper Drive will send the DM1 message every one second, regardless of whether or not there are any active faults, as recommended by the standard. While there are no active DTCs, the Stepper Drive will send the "No Active Faults" message. If a previously inactive DTC becomes active, a DM1 will be sent immediately to reflect this. As soon as the last active DTC goes inactive, it will send a DM1 indicating that there are no more active DTCs.

If there is more than one active DTC at any given time, the regular DM1 message will be sent using a multipacket Broadcast Announce Message (BAM). If the controller receives a request for a DM1 while this is true, it will send the multipacket message to the Requester Address using the Transport Protocol (TP).

> ℹ️ At power up, the DM1 message will not be broadcasted until after a 5 second delay. This is done to prevent any power up or initialization conditions from being flagged as an active error on the network.

When the fault is linked to a DTC, a non-volatile log of the occurrence count (OC) is kept. As soon as the controller detects a new (previously inactive) fault, it will start decrementing the "**Delay Before Sending DM1**" timer for that Diagnostic function block. If the fault has remained present during the delay time, then the controller will set the DTC to active, and will increment the OC in the log. A DM1 will immediately be generated that includes the new DTC. The timer is provided so that intermittent faults do not overwhelm the network as the fault comes and goes, since a DM1 message would be sent every time the fault shows up or goes away.

> ℹ️ Since saving NVM data can affect the performance of the stepper motor, saves of the changes to the OC's in the log will only happen when the motor is not running.

Previously active DTCs (any with a non-zero OC) are available upon request for a **DM2** message. If there is more than one previously active DTC, the multipacket DM2 will be sent to the Requester Address using the Transport Protocol (TP).

Should a **DM3** be requested, the occurrence count of all previously active DTCs will be reset to zero. The OC of currently active DTCs will not be changed.

The Diagnostic function block has a setpoint **"Event Cleared only by DM11."** By default, this is always set to False, which means that as soon as the condition that caused an error flag to be set goes away, the DTC is automatically made Previously Active, and is no longer included in the DM1 message. However, when this setpoint is set to True, even if the flag is cleared, the DTC will not be made inactive, so it will continue to be sent on the DM1 message. Only when a DM11 has been requested will the DTC go inactive. This feature may be useful in a system where a critical fault needs to be clearly identified as having happened, even if the conditions that caused it went away.

In addition to all the active DTCs, another part of the DM1 message is the first byte which reflects the Lamp Status. Each Diagnostic function block has the setpoint **"Lamp Set by Event in DM1"** which determines which lamp will be set in this byte while the DTC is active. The J1939 standard defines the lamps as *'Malfunction', 'Red, Stop', 'Amber, Warning'* or *'Protect'.* By default, the *'Amber, Warning'* lamp is typically the one set by any I/O fault.

By default, every Diagnostic function block has associated with it a proprietary SPN. However, this setpoint **"SPN for Event used in DTC"** is fully configurable by the user should they wish it to reflect a standard SPN define in J1939-71 instead. If the SPN is changed, the OC of the associate error log is automatically reset to zero.

Every Diagnostic function block also has associated with it a default FMI. The only setpoint for the user to change the FMI is **"FMI for Event used in DTC,"** even though some Diagnostic function blocks can have both high and low errors as shown in Table 22. In those cases, the FMI in the setpoint reflect that of the low end condition, and the FMI used by the high fault will be determined per Table 19. If the FMI is changed, the OC of the associate error log is automatically reset to zero.

FMI for Event used in DTC – Low Fault	Corresponding FMI used in DTC – High Fault
FMI=1, Data Valid But Below Normal Operational Range - Most Severe Level	FMI=0, Data Valid But Above Normal Operational Range - Most Severe Level
FMI=4, Voltage Below Normal, Or Shorted To Low Source	FMI=3, Voltage Above Normal, Or Shorted To High Source
FMI=5, Current Below Normal Or Open Circuit	FMI=6, Current Above Normal Or Grounded Circuit
FMI=17, Data Valid But Below Normal Operating Range - Least Severe Level	FMI=15, Data Valid But Above Normal Operating Range - Least Severe Level
FMI=18, Data Valid But Below Normal Operating Range - Moderately Severe Level	FMI=16, Data Valid But Above Normal Operating Range - Moderately Severe Level
FMI=21, Data Drifted Low	FMI=20, Data Drifted High

Table 22 – Low Fault FMI versus High Fault FMI

⚠️	If the FMI used is anything other than one of those in Table 22, then both the low and high faults will be assigned the same FMI. This condition should be avoided, as the log will still used different OC for the two types of faults, even though they will be reported the same in the DTC. It is the user's responsibility to make sure this does not happen.

1.8. CAN Receive Function Block
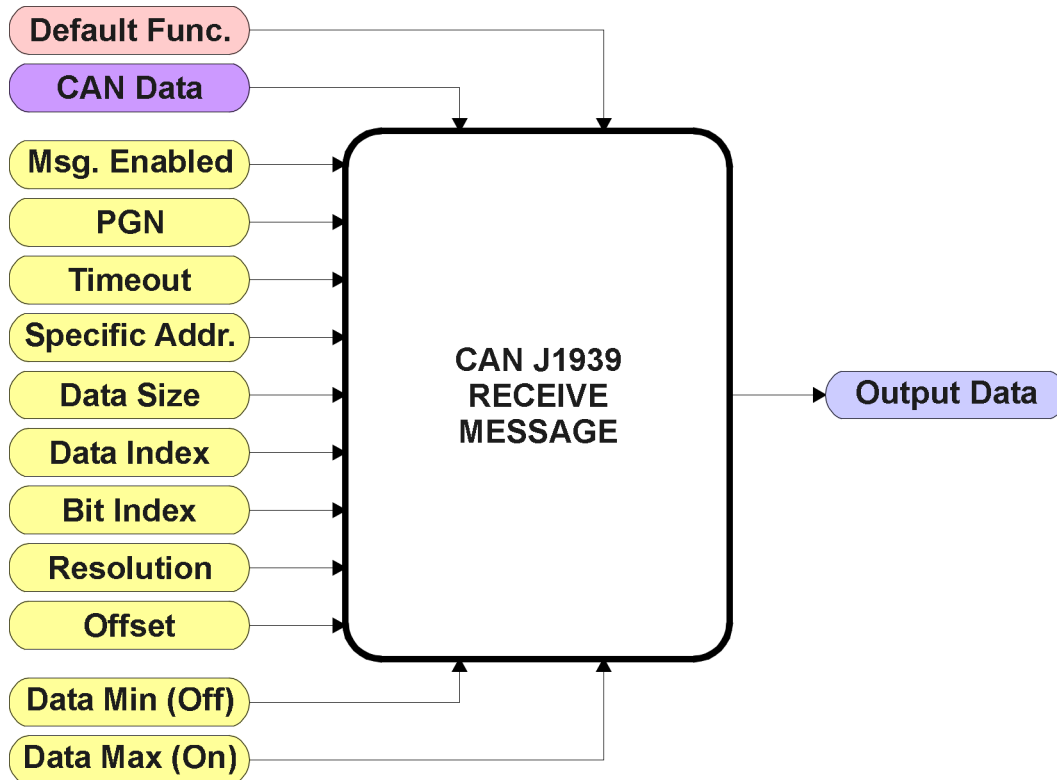
Figure 17 – CAN Receive Function Block

The CAN Receive function block is designed to take any SPN from the J1939 network, and use it as an input to any another function block (i.e. motor or output control).

The **"Received Message Enabled"** is the most important setpoint associated with this function block, and it should be selected first. Changing it will result in other setpoints being enabled/disabled as appropriate. By default ALL received messages are enabled.

> ⚠️ Since the drive can support up to 15 received messages, it can get confusing as to what message performs what function. Therefore, the intended **"Default Receive Function"** is available as a read-only value for each message. Change functions with CAUTION!

Once a message has been enabled, a Lost Communication fault will be flagged if that message is not received off of the bus within the **"Receive Message Timeout"** period. This will trigger a motor shutdown as described in Section 1.7. In order to avoid timeouts on a heavily saturated network, it is recommended to set it at least three times larger than the expected update rate. To disable the timeout feature, simply set this value to zero, in which case the received message will never trigger a Lost Communication fault.

By default, all control messages are expected to be sent to the drive on Proprietary B PGNs. However, should a PDU1 message be selected, the drive can be setup to receive it from any ECU by setting the **"Specific Address that sends the PGN"** to the Global Address (0xFF). If a specific address is selected instead, then any other ECU sending data on the PGN will be ignored.

The **"Receive Data Size"**, **"Receive Data Index in Array (LSB)"**, **"Receive Bit Index in Byte (LSB)"**, **"Receive Resolution"** and **"Receive Offset"** can all be use to map any SPN supported by the J1939 standard to the output data of the Received function block. The defaults used by the drive are all for proprietary SPNs, and are defined in detail in Section 3.3.

Note: Output Data = CAN Data * Resolution + Offset

As mentioned earlier, a CAN Receive function block can be selected as the source of the control input for the output function blocks. When this is the case, the **"Received Data Min (Off Threshold)"** and **"Received Data Max (On Threshold)"** setpoints become the minimum and maximum values of the X-axis used in the linear calculations. As the names imply, they are also used as the On/Off thresholds for digital input types. These values are in whatever units the output data is AFTER the resolution and offset is applied to the CAN data.

The Stepper Drive supports up to 15 unique CAN Receive Messages. By default, each message is pre-configured to read a particular type of data. The details are outlined in Section 3.3, and the default list is shown in the table below.

Block #	Default Receive Data
1	Command Motor Mode
2	Command Direction
3	Command Step Rate
4	Command Max. Phase Current
5	Command Number of Steps
6	Set Step Type
7	Set Percent Fast Decay
8	Set Acceleration Table
9	Put Drive to Sleep
10	Reset Motor Drive
11	E-Stop Motor Drive
12	Command Digital Output
13	Save Data (Send Once)
14	PID Target Input Data
15	PID Feedback Input Data

Table 23 – Default CAN Receive Messages

1.9. CAN Transmit Function Block
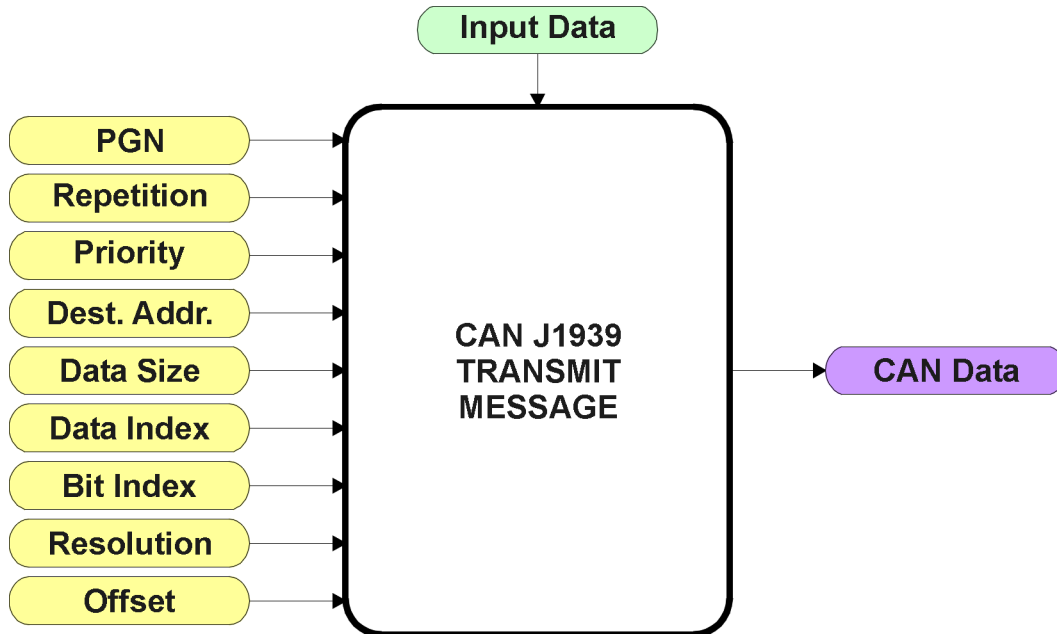
Figure 18 – CAN Transmit Function Block

The CAN Transmit function block is used to send any output from another function block (i.e. input, status or feedback signals) to the J1939 network.

Normally, to disable a transmit message, the **"Transmit Repetition Rate"** is set to zero. However, should message share its Parameter Group Number (PGN) with another message, this is not necessarily true. In the case where multiple messages share the same **"Transmit PGN"**, the repetition rate selected in the message with the LOWEST number will be used for ALL the messages that use that PGN.

By default, all messages are sent on Proprietary B PGNs as broadcast messages. The default settings do 'bundle' multiple messages onto a PGN, as outlined in Section 3. If all of the data is not necessary, disable the entire message by setting the lowest channel using that PGN to zero. If some of the data is not necessary, simply change the PGN of the superfluous channel(s) to an unused value in the Proprietary B range.

> At power up, transmitted message will not be broadcasted until after a 1 second delay. This is done to prevent any power up or initialization conditions from creating problems on the network.

Since the defaults are PropB messages, the **"Transmit Message Priority"** is always initialized to 6 (low priority) and the **"Destination Address (for PDU1)"** setpoint is not used. This setpoint is only valid when a PDU1 PGN has been select, and it can be set either to the Global Address (0xFF) for broadcasts, or sent to a specific address as setup by the user.

The **"Transmit Data Size"**, **"Transmit Data Index in Array (LSB)"**, **"Transmit Bit Index in Byte (LSB)"**, **"Transmit Resolution"** and **"Transmit Offset"** can all be use to map the data to any SPN supported by the J1939 standard. The defaults used by the drive are all for proprietary SPNs, and are defined in detail in Section 3.4.

Note: CAN Data = (Input Data – Offset)/Resolution

The Stepper Drive supports up to 10 unique CAN Transmit Messages, all of which can be programmed to send any available data to the CAN network. By default, each message is pre-configured to send a particular type of data. The details are outlined in Section 3.4, and the default list is shown in the table below.

Block #	Default Transmit Data
1	Averaged Stepper Phase 1 Current
2	Averaged Stepper Phase 2 Current
3	Stepper Variable Feedback, Motor Drive Mode
4	Stepper Variable Feedback, Rotation Direction
5	Stepper Variable Feedback, Step Rate
6	Input Measured Value, Voltage/Current Input 1
7	Input Measured Value, Frequency/PWM Input 1
8	Digital Output Feedback State
9	Power Supply Measured
10	Spare Message, Not Used by Default

Table 24 – Default CAN Transmit Messages

1.10.   Available Control/Input Sources

Throughout Section 1, there have been numerous mentions of 'mapping' an output of one function block to a control or input of another. Any time an input is programmable, as shown by the green bubbles in the figures above, it consists of two setpoints. One is the **"[Name] Source"** and the other the **"[Name] Number."** Together, these two setpoints uniquely select how the I/O of the various function blocks is linked together in the custom application.

Below is the full list of the programmable inputs supported by the Stepper Drive

Function Block	Input Name
Stepper Drive Controls	Motor Mode
	Direction
	Step Rate
	Max. Phase Current
	Number of Steps
	Step Type
	Percent Fast Decay
	Acceleration Table
	Put Drive to Sleep
	Reset Motor Drive
	E-Stop Motor Drive
	Save Data to Defaults
PID Control Function	PID Target Command
	PID Feedback Input
Digital Output	Control
CAN Transmit X (where X = 1 to 10)	Transmit Data

Table 25 – Programmable Inputs

For each of the inputs listed in Table 25, the options for the Source and Number are listed below. Note, not all options would make sense for any particular input, and it is up to the user to program the controller in a logical and functional fashion.

Sources	Range	Notes
Control Source Not Used	N/A	When this is selected, it disables all other setpoints associated with the input in question.
Received CAN J1939 Message	1 to 15	The CAN Receive Messages each have a unique default assigned to them as outlined in Section 1.8
Input Measured Value	1 or 2	1 = Voltage/Current Input 1 2 = Frequency/PWM Input 1
Digital Input Read State	1 or 2	1 = Digital Input 1 (Analog Input) 2 = Digital Input 2 (Frequency Input)
Digital Output Feedback State	N/A	Can be mapped to a CAN Txd Message
Averaged Stepper Phase Current	1 or 2	Intended for error detection feedback only. **Not a reflection of the true RMS value.** Can be mapped to a CAN Txd Message
Power Supply Measured	N/A	Input to Power Supply Diagnostics Can be mapped to a CAN Txd
Stepper Variable Feedback	1 to 12	1 = Save Operating Data 2 = Put Drive to Sleep 3 = Reset Motor 4 = E-Stop Motor 5 = Motor Drive Mode 6 = Drive Step Type 7 = Rotation Direction 8 = Selected Acc. Table 9 = Percent Fast Decay 10 = Maximum Phase Current 11 = Number Of Steps 12 = Step Rate *These are intended primarily to be mapped to a CAN Txd Message, although they could be a control for the digital output (i.e. reflect direction)
PID Function Block Control	1 to 4	1 = PID Setpoint Value 2 = PID Commanded Motor Mode 3 = PID Commanded Direction 4 = PID Commanded Step Rate 5 = PID Delta Out of Range
Reflect When a Fault is Detected	N/A	Only to be used as a Digital Output X, Control Source

Table 26 – Input Sources and Numbers

2. INSTALLATION INSTRUCTIONS

2.1. Dimensions and Pinout
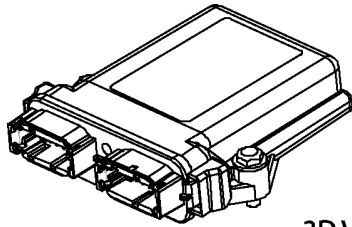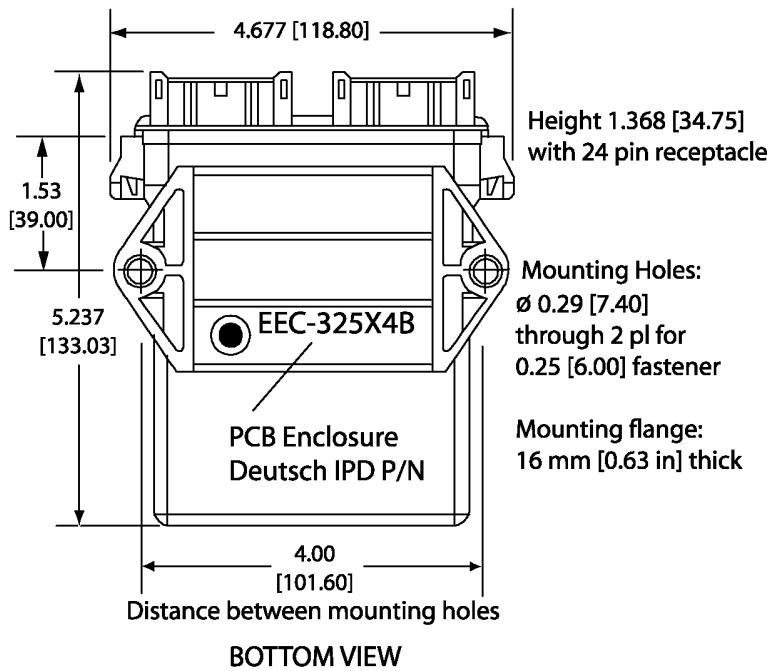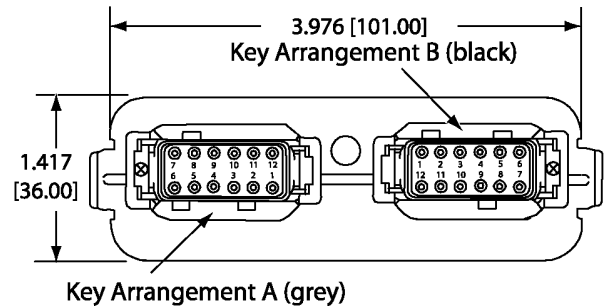
Figure 19 – Dimension Drawing

Grey Connector		Black Connector	
Pin #	Function	Pin #	Function
1	No Connection	1	Phase 1, A+
2	CAN Shield	2	Phase 2, B+
3	No Connection	3	Digital Output + (+Vcc)
4	Frequency GND	4	Frame GND (external connection)
5	+5V Reference GND (AGND)	5	Battery -
6	Analog GND (AGND)	6	Battery +
7	Analog Input (Voltage/Current 1 or Din1)	7	Battery +
8	+5V Reference (up to 10mA)	8	Battery -
9	Frequency Input (Frequency/PWM 1 or Din2)	9	Frame GND (external connection)
10	No Connection	10	Digital Output – (up to 5A)
11	CAN Low	11	Phase 2, B-
12	CAN High	12	Phase 1, A-

Table 27 – Connector Pinout

3. OVERVIEW OF J1939 FEATURES

The software was designed to provide flexibility to the user with respect to messages sent to and from the ECU by providing:

- Configurable ECU Instance in the NAME (to allow multiple ECUs on the same network)
- Configurable Receive PGN and SPN Parameters
- Configurable Transmit PGN and SPN Parameters
- Sending DM1 Diagnostic Message Parameters
- Diagnostic Log, maintained in non-volatile memory, for sending DM2 messages

3.1. Introduction To Supported Messages
The ECU is compliant with the standard SAE J1939, and supports the following PGNs

From J1939-21 - Data Link Layer
- Request   59904 ($00EA00)
- Acknowledgment   59392 ($00E800)
- Transport Protocol – Connection Management   60416 ($00EC00)
- Transport Protocol – Data Transfer Message   60160 ($00EB00)
- PropB Receive, Default Motor Command Message   65280 ($00FF00)
- PropB Receive, Default Motor Setup and Control Message   65296 ($00FF10)
- PropB Receive, Default PID Target and Feedback Messages   65408 ($00FF80)
- PropB Transmit, Default Motor Feedback Message   65360 ($00FF50)
- PropB Transmit, Default I/O Feedback Message   65365 ($00FF55)

Note: Any Proprietary B PGN in the range 65280 to 65535 ($00FF00 to $00FFFF) can be selected
Note: The Proprietary A PGN 61184 ($00EF00) can also be selected for any of the messages

From J1939-73 - Diagnostics
- DM1 – Active Diagnostic Trouble Codes   65226 ($00FECA)
- DM2 – Previously Active Diagnostic Trouble Codes   65227 ($00FECB)
- DM3 – Diagnostic Data Clear/Reset for Previously Active DTCs   65228 ($00FECC)
- DM11 - Diagnostic Data Clear/Reset for Active DTCs   65235 ($00FED3)
- DM14 – Memory Access Request   55552 ($00D900)
- DM15 – Memory Access Response   55296 ($00D800)
- DM16 – Binary Data Transfer   55040 ($00D700)

From J1939-81 - Network Management
- Address Claimed/Cannot Claim   60928 ($00EE00)
- Commanded Address   65240 ($00FED8)

From J1939-71 – Vehicle Application Layer
- Software Identification   65242 ($00FEDA)

None of the application layer PGNs is supported as part of the default configurations, but they can be selected as desired for either transmit or received function blocks.

Setpoints are accessed using standard Memory Access Protocol (MAP) with proprietary addresses. The Axiomatic Electronic Assistant (EA) allows for quick and easy configuration of the unit over the CAN network.

3.2. NAME, Address and Software ID

The NAME, Address and Software ID have the following default configurations.

J1939 NAME
The Stepper Drive ECU has the following defaults for the J1939 NAME. The user should refer to the SAE J1939/81 standard for more information on these parameters and their ranges.

Arbitrary Address Capable	Yes
Industry Group	0, Global
Vehicle System Instance	0
Vehicle System	0, Non-specific system
Function	132, Axiomatic Motor Control
Function Instance	28, Axiomatic 2A Stepper Motor Drive
ECU Instance	**0, First Instance**
Manufacture Code	162, Axiomatic Technologies Corporation
Identity Number	Variable, based on ECU Serial Number

The ECU Instance is a configurable setpoint associated with the NAME. Changing this value will allow multiple ECUs of this type to be distinguishable by other ECUs (including the Axiomatic Electronic Assistant) when they are all connected on the same network.

ECU Address
The default value of this setpoint is 200 (0xC8), which is a Motor Controller preferred address as set by the SAE in J1939 tables B3 to B7. The Axiomatic EA supports the selection of any address between 0 to 253, and *it is the user's responsibility to select an address that complies with the standard*. The user must also be aware that since the unit is arbitrary address capable, if another ECU with a higher priority NAME contends for the selected address, the Stepper Drive will continue select the next highest address until it find one that it can claim. See J1939/81 for more details about address claiming.

Software Identifier

PGN 65242	Software Identification		- SOFT
Transmission Repetition Rate:	On request		
Data Length:	Variable		
Extended Data Page:	0		
Data Page:	0		
PDU Format:	254		
PDU Specific:	218 PGN Supporting Information:		
Default Priority:	6		
Parameter Group Number:	65242 (0xFEDA)		

Start Position	Length	Parameter Name	SPN
1	1 Byte	Number of software identification fields	965
2-n	Variable	Software identification(s), Delimiter (ASCII "*")	234

For the Stepper Drive ECU, Byte 1 is set to 5, and the identification fields are as follows

(Part Number)*(Version)*(Date)*(Owner)*(Description)

The Axiomatic EA shows all this information in "General ECU Information", as shown below:

Parameter	Value	Description
► ECU Part Number	AX100700	
► ECU Serial Number	0021710039	
┌ ECU J1939 NAME		PGN 60928. 64-bit ECU Identifi
► Arbitrary Address Capable	0X01	Yes
► Industry Group	0X00	Global
► Vehicle System Instance	0X00	
► Vehicle System	0X00	Non-specific system
► Reserved	0X00	
► Function	0X84	Axiomatic Motor Controller
► Function Instance	0X1C	
► ECU Instance	0X00	#1 - First Instance
► Manufacturer Code	0X0A2	Axiomatic Technologies
► Identity Number	0X1EB37E	Unique ECU network ID numbe
► ECU Address	0XC8	Reserved for future assignmer
┌ Software ID		PGN 65242 -SOFT
► Field #1	AX100700	
► Field #2	V2.1.0	
► Field #3	MAY 2012	
► Field #4	Axiomatic Technologies	
► Field #5	Stepper Motor Drive, 2A	

3.3. CAN Receive Message Defaults

This section outlines the **default** settings of the stepper drive CAN receive channels, used as inputs to the various function blocks supported by this ECU. Recall, however, that this is a fully programmable unit, such that all these SPNs can be received on different PGNs if so desired.

In all the messages shown below, none of the received values have an SPN assigned to them, as this drive only uses the SPNs for diagnostic trouble codes. To have the drive react to a DTC sent by another ECU on the network on a DM1, use the DTC React Function block instead.

The "Motor Control Messages" has the following default configuration.

```
PGN 65280              Motor Command Messages


Transmission Repetition:          100ms (default, configurable)
Data Length:                      8
Data Page:                        0
PDU Format:                       254
PDU Specific:                     GE    PGN Supporting Information:
Default Priority:                 6
Parameter Group Number:           65280 (0xFF00)
```

Start Position	Length	Parameter Name	SPN
1	1 byte	Command Motor Mode	N/A
2	1 byte	Command Direction	N/A
3-4	2 bytes	Command Step Rate	N/A
5-6	2 bytes	Command Max. Phase Current	N/A
7-8	2 bytes	Command Number of Steps	N/A

The "Motor Setup and Control Message" has the following default configuration.

```
PGN 65296              Motor Setup and Control Messages


Transmission Repetition:          0s (only sent once, or on request by default, configurable)
Data Length:                      8
Data Page:                        0
PDU Format:                       254
PDU Specific:                     GE    PGN Supporting Information:
Default Priority:                 6
Parameter Group Number:           65296 (0xFF10)
```

Start Position	Length	Parameter Name	SPN
1	1 byte	Set Step Type	N/A
2	1 byte	Set Percent Fast Decay	N/A
3	1 byte	Set Acceleration Table	N/A
4	1 byte	Put Drive to Sleep	N/A
5	1 byte	Reset Motor Drive	N/A
6	1 byte	E-Step Motor Signal	N/A
7	1 byte	Command Digital Output	N/A
8	1 byte	Save Data (Send Once)	N/A

Command Motor Mode

Default value used when a *'Received CAN J1939 Message'* is selected as the control source for the motor mode.

    00   Motor Disabled
    01   Motor Stopped
    02   Motor Running

Data Length: 1 byte
Resolution: 1 mode/bit, 0 offset
Data Range: 0 to 2   (Note: valid data >2 will be truncated to 2)
Type: Input
Suspect Parameter Number: N/A
Parameter Group Number: 65280

Command Direction

Default value used when a *'Received CAN J1939 Message'* is selected the control source for the direction of motor rotation.

    00   Counter Clockwise
    01   Clockwise

Data Length: 1 byte
Resolution: 1 direction/bit, 0 offset
Data Range: 0 or 1   (Note: valid data >1 will be truncated to 1)
Type: Input
Suspect Parameter Number: N/A
Parameter Group Number: 65280

Command Step Rate

Default value used when a *'Received CAN J1939 Message'* is selected the control source for the step rate applied to the stepper motor.

Data Length: 2 bytes
Resolution: 1 [StepUnit]/bit, 0 offset
Data Range: 0 to 64255 [StepUnit], default 1 to 50
Type: Input
Suspect Parameter Number: N/A
Parameter Group Number: 65280

Command Max. Phase Current

Default value used when a *'Received CAN J1939 Message'* is selected the control source for the maximum current setting applied to the driver chip.

Data Length: 2 bytes
Resolution: 1 mA/bit, 0 offset
Data Range: 0 to 64255 [mA], default 100 to 2000
Type: Input
Suspect Parameter Number: N/A
Parameter Group Number: 65280

Command Number of Steps

Default value used when a *'Received CAN J1939 Message'* is selected the control source for the number of steps the motor will run before holding or stopping.

Data Length: 2 bytes
Resolution: 1 step/bit, 0 offset
Data Range: 0 to 64255 [Step], default 0 to 1000
Type: Input
Suspect Parameter Number: N/A
Parameter Group Number: 65280

Set Step Type

Default value used when a *'Received CAN J1939 Message'* is selected the control source for the selection of the micro-stepping type applied at the phase current outputs (see Figures 5 and 6).

    00   Full Step
    01   Half Step
    02   Quarter Step
    03   Sixteenth Step

Data Length: 1 byte
Resolution: 1 type/bit, 0 offset
Data Range: 0 to 3   (Note: valid data >3 will be truncated to 3)
Type: Input
Suspect Parameter Number: N/A
Parameter Group Number: 65296

Set Percent Fast Decay

Default value used when a *'Received CAN J1939 Message'* is selected the control source for the PFD used by the driver chip, A3979.

    00   Slow
    01   Fast
    02   Mixed

Data Length: 1 byte
Resolution: 1 PFD/bit, 0 offset
Data Range: 0 to 2   (Note: valid data >2 will be truncated to 2)
Type: Input
Suspect Parameter Number: N/A
Parameter Group Number: 65296

Set Acceleration Table

Default value used when a *'Received CAN J1939 Message'* is selected the control source for the selection of the acceleration/deceleration profile used when the motor starts or stops.

    00   NoProfile
    0X   ProfileX, where X = 1 to 5

Data Length: 1 byte
Resolution: 1 table/bit, 0 offset
Data Range: 0 to 5   (Note: valid data >5 will be truncated to 5)
Type: Input
Suspect Parameter Number: N/A
Parameter Group Number: 65296

Put Drive to Sleep

Default value used when a *'Received CAN J1939 Message'* is selected the control source for putting the driver chip, A3979, to sleep

    00   False (Drive not asleep)

    01   True   (Drive in sleep mode)

Data Length:	1 byte
Resolution:	1 state/bit, 0 offset
Data Range:	0 or 1 (Note: valid data >1 will be truncated to 1)
Type:	Input
Suspect Parameter Number:	N/A
Parameter Group Number:	65296

Reset Motor Drive

Default value used when a *'Received CAN J1939 Message'* is selected the control source for resetting the driver chip, A3979, to the Home position when microstepping

    00   False (Drive not reset)

    01   True   (Drive reset to home position)

Data Length:	1 byte
Resolution:	1 state/bit, 0 offset
Data Range:	0 or 1 (Note: valid data >1 will be truncated to 1)
Type:	Input
Suspect Parameter Number:	N/A
Parameter Group Number:	65296

E-Stop Motor Signal

Default value used when a *'Received CAN J1939 Message'* is selected the control source for emergency stopping the motor

    00   False (Drive not in E-Stop)

    01   True   (E-Stop motor)

Data Length:	1 byte
Resolution:	1 state/bit, 0 offset
Data Range:	0 or 1 (Note: valid data >1 will be truncated to 1)
Type:	Input
Suspect Parameter Number:	N/A
Parameter Group Number:	65296

Command Digital Output

Default value used when a *'Received CAN J1939 Message'* is selected the control source for the digital output on the motor drive

    00   OFF   (output commanded OFF)

    01   ON   (output commanded ON)

Data Length:	1 byte
Resolution:	1 state/bit, 0 offset
Data Range:	0 or 1 (Note: valid data >1 will be truncated to 1)
Type:	Input
Suspect Parameter Number:	N/A
Parameter Group Number:	65296

Save Data (Sent Once)
Default value used when a *'Received CAN J1939 Message'* is selected the control source for saving the current settings of the motor variable to non-volatile memory to be used as the power-up and reset defaults. The flag to save is only set on the rising edge of the signal, i.e. a transition from False to True. The actual save operation will only take place once the motor is not running.

    00   False (Do not save current settings)
    01   True   (Save current settings)

Data Length:	1 byte
Resolution:	1 state/bit, 0 offset
Data Range:	0 or 1 (Note: valid data >1 will be truncated to 1)
Type:	Input
Suspect Parameter Number:	N/A
Parameter Group Number:	65296

The "PID Target and Feedback Messages" has the following default configuration.

```
PGN 65408          PID Target and Feedback Messages


Transmission Repetition:      0s (default, configurable)
Data Length:                  8
Data Page:                    0
PDU Format:                   254
PDU Specific:                 GE     PGN Supporting Information:
Default Priority:             6
Parameter Group Number:       65408 (0xFF80)


Start Position   Length    Parameter Name                    SPN
1                1 byte    PID Target Input Data              N/A
2                1 byte    PID Feedback Input Data            N/A
```

PID Target Input Data
Default value used when a *'Received CAN J1939 Message'* is selected the control source for the PID control function's target value.

Data Length:	1 byte
Resolution:	0.4 %/bit, 0 offset
Data Range:	0 to 250, default 0% to 100%
Type:	Input
Suspect Parameter Number:	N/A
Parameter Group Number:	65408

PID Feedback Input Data
Default value used when a *'Received CAN J1939 Message'* is selected the control source for the PID control function's feedback value.

Data Length:	1 byte
Resolution:	0.4 %/bit, 0 offset
Data Range:	0 to 250, default 0% to 100%
Type:	Input
Suspect Parameter Number:	N/A
Parameter Group Number:	65408

3.4. CAN Transmit Message Defaults

This section outlines the **default** settings of the stepper drive CAN transmissions. Recall, however, that this is a fully programmable unit, such that all these SPNs can be sent on different PGNs if so desired.

In all the messages shown below, not all the transmitted values have an SPN assigned to them, as this drive only uses the SPNs for diagnostic trouble codes. If the SPN is shown as N/A, this means that the associated value cannot be used to generate DTCs.

The "Motor Feedback Message" has the following default configuration.

PGN 65360		Motor Feedback Message	
Transmission Repetition:		1s (default, configurable)	
Data Length:		8	
Data Page:		0	
PDU Format:		254	
PDU Specific:		GE PGN Supporting Information:	
Default Priority:		6	
Parameter Group Number:		65360 (0xFF50)	
Start Position	Length	Parameter Name	SPN
1-2	2 bytes	Averaged Stepper Phase 1 Current	520193 ($7F001)
3-4	2 bytes	Averaged Stepper Phase 2 Current	520194 ($7F002)
5	1 byte	Stepper Feedback, Motor Drive Mode	N/A
6	1 byte	Stepper Feedback, Rotation Direction	N/A
7-8	2 bytes	Stepper Feedback, Step Rate	N/A

The "I/O Feedback Message" has the following default configuration.

PGN 65365		I/O Feedback Message	
Transmission Repetition:		1s (default, configurable)	
Data Length:		8	
Data Page:		0	
PDU Format:		254	
PDU Specific:		GE PGN Supporting Information:	
Default Priority:		6	
Parameter Group Number:		65365 (0xFF55)	
Start Position	Length	Parameter Name	SPN
1-2	2 byte	Input Measured Value, Voltage/Current 1	520208 ($7F010)
3-4	2 byte	Input Measured Value, Frequency/PWM 1	520448 ($7F100)
5	1 byte	Digital Output Feedback State	520195 ($7F003)
6	1 byte	Not Used (set to 0xFF)	N/A
7-8	2 byte	Power Supply Measured	524273 ($7FFF1)

Averaged Stepper Phase X Current, where X = 1 or 2

Reflects the averaged current measured from the current sense resistor on the corresponding phase. This is NOT a reflection of the true RMS current value, simply a feedback signal that can be used for diagnostic purposes to detect an open-circuit on the phase.

Data Length:	2 bytes
Resolution:	1 mA/bit, 0 offset
Type:	Measured
Suspect Parameter Number:	520193 (Phase 1) or 520194 (Phase 2)
Parameter Group Number:	65360

Stepper Feedback, Motor Drive Mode

Reflects the actual mode the stepper motor is operating in at the time. Typically, this signal will reflect the commanded mode, but it can also reflect when the motor is Stepping or Holding, or if it has been disabled because of an E-Stop signal or diagnostics.

00	Motor Disabled
01	Motor Stopped
02	Motor Running
03	Motor Stepping
04	Motor Holding

Data Length:	1 byte
Resolution:	1 mode/bit, 0 offset
Type:	Status
Suspect Parameter Number:	N/A
Parameter Group Number:	65360

Stepper Feedback, Rotation Direction

Reflects the direction of rotation of the motor. This feedback is primarily intended for use in conjunction with in the PID Control method of driving the motor. In CAN commanded mode, this byte will simply reflect the commanded direction that has been set.

00	Counter Clockwise
01	Clockwise

Data Length:	1 byte
Resolution:	1 direction/bit, 0 offset
Type:	Status
Suspect Parameter Number:	N/A
Parameter Group Number:	65360

Stepper Feedback, Step Rate

Reflects the step rate of the motor. This feedback is primarily intended for use in conjunction with in the PID Control method of driving the motor. In CAN commanded mode, this byte will simply reflect the commanded step rate that has been set. The data will be in **"Step Rate Data Unit"**

Data Length:	2 bytes
Resolution:	1 [StepUnit]/bit, 0 offset
Type:	Calculated
Suspect Parameter Number:	N/A
Parameter Group Number:	65360

Input Measured Value, Voltage/Current 1
This value reflects the measured value of the Analog Input
Data Length: 2 byte
Resolution: 0.001 [Unit]/bit, 0 offset
Data Range: 0 to 5V (5,000mV) or 0 to 20mA   (20,000uA)
Type: Measured
Suspect Parameter Number: 520208   (0x7F010, a proprietary SPN)
Parameter Group Number: 65365

Input Measured Value, Frequency/PWM 1
This value reflects the measured value of the Frequency Input
Data Length: 2 byte
Resolution: 0.1 [Unit]/bit, 0 offset
Data Range: 0 to 10,000Hz (max 100,000) or 0 to 100% (max 1,000)
Type: Measured
Suspect Parameter Number: 520448   (0x7F100, a proprietary SPN)
Parameter Group Number: 65365

Digital Output Feedback State
        00   Output OFF
        01   Output ON
        02   Output Error
        03   Output Not Applicable (i.e. control source disabled)
Data Length: 1 byte
Resolution: 1 state/bit, 0 offset
Type: Measured
Suspect Parameter Number: 520195   (0x7F003 a proprietary SPN)
Parameter Group Number: 65365

Power Supply Measured
This value reflects the measured supply voltage powering the stepper drive.
Data Length: 2 byte
Resolution: 0.01 V/bit, 0 offset
Data Range: 0 to 642.55 V
Type: Measured
Suspect Parameter Number: 524273   (0x7FFF1, a proprietary SPN)
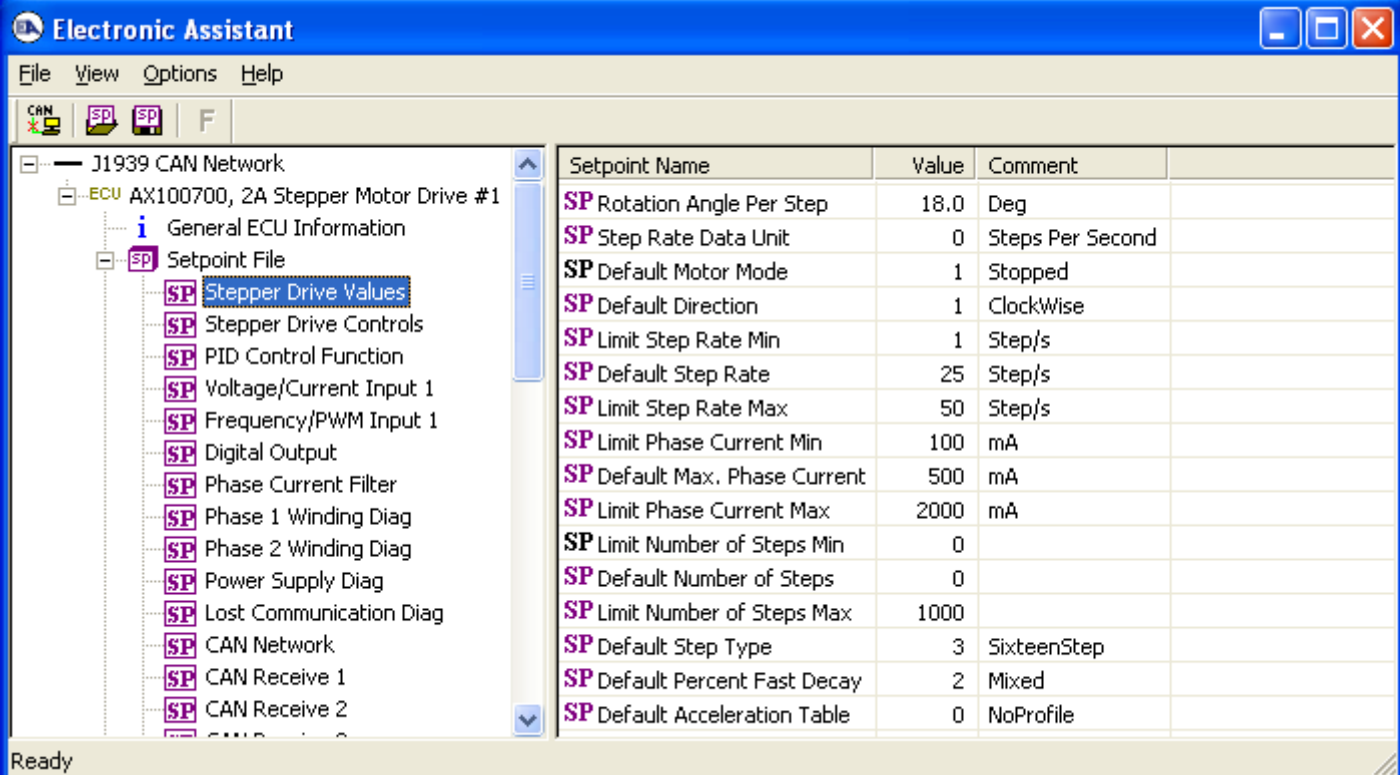Parameter Group Number: 65365

4. ECU SETPOINTS ACCESSED WITH THE EA

Many setpoints have been referenced throughout this manual. This section describes in detail each setpoint, and their defaults and ranges. For more information on how each setpoint is used by the Stepper Drive, refer to the relevant section of the User Manual.

4.1. Stepper Drive Values Setpoints

This function block defines the default values used by the driver after a power-up or reset cycle, as well as allowing the user to set the limits for the linear controls.

The Motor Drive function blocks are defined in Section 1.2. Please refer there for detailed information about how all these setpoints are used.

Screen Capture of Default Stepper Drive Values Setpoints

Name	Range	Default	Notes
Rotation Angle Per Step	0.1 to 360.0º	18.0º	Per Stepper Motor Datasheet
Step Rate Data Units	Drop List	0, Steps Per Second	0 = Steps Per Second 1 = Rotations Per Minute
Default Motor Mode	N/A	1, Stopped	Read-Only Parameter See Table 2
Default Direction	Drop List	1, ClockWise	See Table 3
Limit Step Rate Min	1 to Default	1 [Unit]	Unit = Step Rate Data Units
Default Step Rate	Min to Max	25 [Step/s]	Unit = Step Rate Data Units
Limit Step Rate Max	Default to Limit*	50 [Step/s]	Unit = Step Rate Data Units
Limit Phase Current Min	100mA to Default	100 mA	
Default Max. Phase Current	Min to Max	500 mA	See Figures 5 and 6
Limit Phase Current Max	Default to 2000mA	2000 mA	
Limit Number of Steps Min	N/A	0	Read-Only Parameter
Default Number of Steps	0 to Max	0	
Limit Number of Steps Max	Default to 10,000,000	1000	
Default Step Type	Drop List	3, SixteenStep	See Table 4
Default Percent Fast Decay	Drop List	2, Mixed	See Table 5
Default Acceleration Table	Drop List	0, NoProfile	See Table 6

Step Rate maximum limit is dependent on Step Type selected. See table below

Step Type	Full Step	Half Step	Quarter Step	Sixteenth Step
Maximum [Step/s]	12,500	6,250	3,125	780
Maximum [RPM]*	37,500	18,750	9,375	2,340

*Where the RPM limit is calculated by

$$Max = (Max[Step/s]) \times (\textbf{“Rotation Angle Per Step”}[º/Step]) \times (60[s/min]) / (360[º/Rotation])$$

The driver chip A3979 will microstep once on a low to high transition on the STEP input. Since this control signal is driven by a 10us timer that will toggle the drive signal twice per microstep (low to high, and high to low), resolution will be lost at higher speeds. For high step rates, a more crude **“Step Type”** should be selected to get better resolution. At lower speeds, the *‘SixteenStep’* type should be used for optimum performance from the motor.
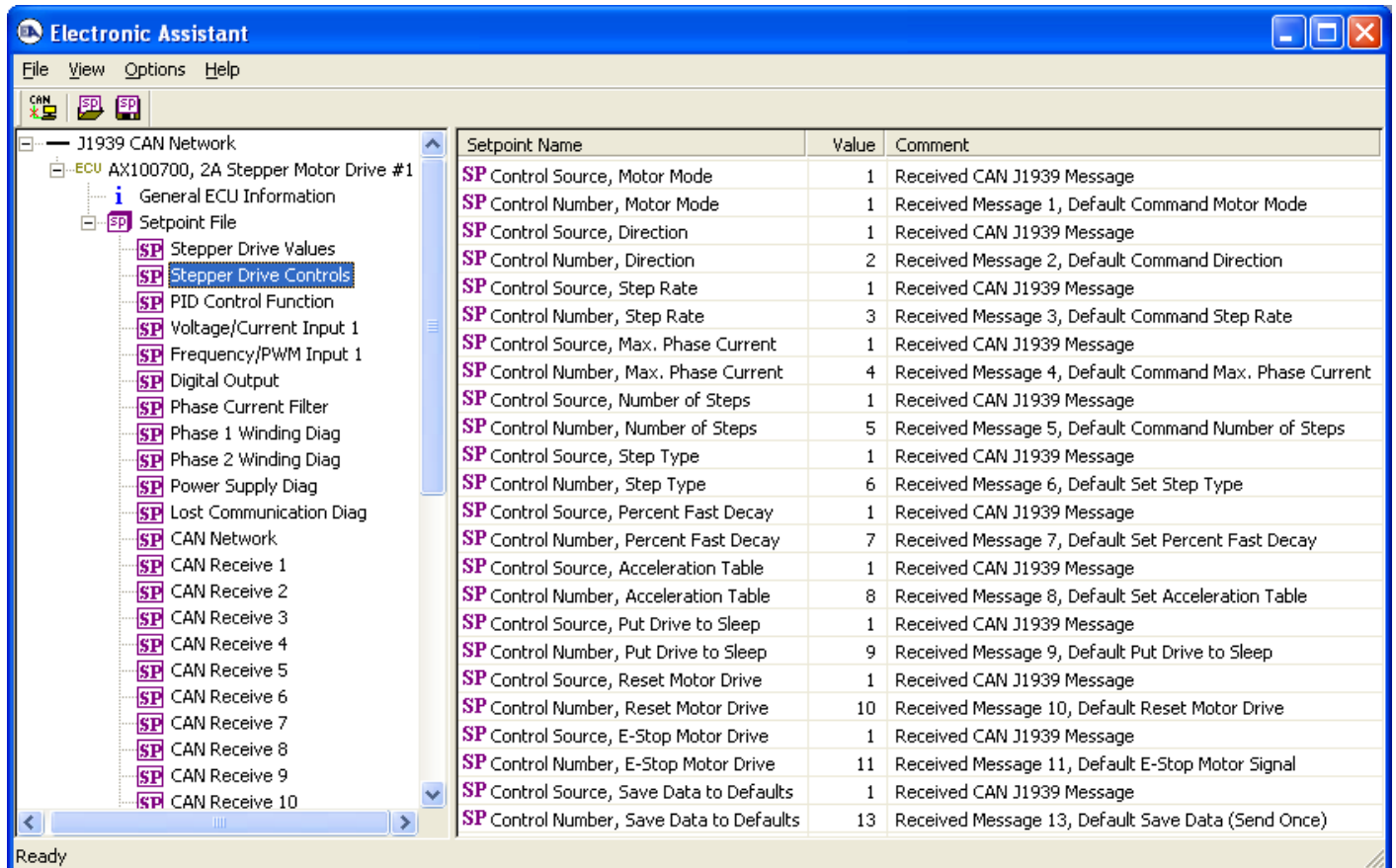
An example of how resolution is lost at higher speeds is shown in the table on the following page.

Timer Reload	MicroStep/Sec	1/16 Step Rate	1/4 Step Rate	Half Step Rate	Full Step Rate
1	50000	781.25	3125	6250	12500
2	25000	390.63	1562.5	3125	6250
3	16666.70	260.42	1041.67	2083.34	4166.68
4	12500.00	195.31	781.25	1562.50	3125.00
5	10000.00	156.25	625.00	1250.00	2500.00
6	8333.30	130.21	520.83	1041.66	2083.33
7	7142.90	111.61	446.43	892.86	1785.73
8	6250.00	97.66	390.63	781.25	1562.50
9	5555.60	86.81	347.23	694.45	1388.90
10	5000.00	78.13	312.50	625.00	1250.00
11	4545.50	71.02	284.09	568.19	1136.38
12	4166.70	65.10	260.42	520.84	1041.68
13	3846.20	60.10	240.39	480.78	961.55
14	3571.40	55.80	223.21	446.43	892.85
15	3333.30	52.08	208.33	416.66	833.33
16	3125.00	48.83	195.31	390.63	781.25
17	2941.20	45.96	183.83	367.65	735.30
18	2777.80	43.40	173.61	347.23	694.45
19	2631.60	41.12	164.48	328.95	657.90
20	2500.00	39.06	156.25	312.50	625.00
21	2381.00	37.20	148.81	297.63	595.25
22	2272.70	35.51	142.04	284.09	568.18
23	2173.90	33.97	135.87	271.74	543.48
24	2083.30	32.55	130.21	260.41	520.83
25	2000.00	31.25	125.00	250.00	500.00
26	1923.10	30.05	120.19	240.39	480.78
27	1851.90	28.94	115.74	231.49	462.98
28	1785.70	27.90	111.61	223.21	446.43
29	1724.10	26.94	107.76	215.51	431.03
30	1666.70	26.04	104.17	208.34	416.68
31	1612.90	25.20	100.81	201.61	403.23
32	1562.50	24.41	97.66	195.31	390.63
33	1515.20	23.68	94.70	189.40	378.80
34	1470.60	22.98	91.91	183.83	367.65
35	1428.60	22.32	89.29	178.58	357.15
36	1388.90	21.70	86.81	173.61	347.23
37	1351.40	21.12	84.46	168.93	337.85
38	1315.80	20.56	82.24	164.48	328.95
39	1282.10	20.03	80.13	160.26	320.53
40	1250.00	19.53	78.13	156.25	312.50
41	1219.50	19.05	76.22	152.44	304.88
42	1190.50	18.60	74.41	148.81	297.63
43	1162.80	18.17	72.68	145.35	290.70
44	1136.40	17.76	71.03	142.05	284.10
45	1111.10	17.36	69.44	138.89	277.78
46	1087.00	16.98	67.94	135.88	271.75
47	1063.80	16.62	66.49	132.98	265.95
48	1041.70	16.28	65.11	130.21	260.43
49	1020.40	15.94	63.78	127.55	255.10
50	1000.00	15.63	62.50	125.00	250.00

4.2. Stepper Drive Controls Setpoints

This function block defines the control sources and numbers for all the command and setup parameters associated with the stepper motor drive.

The Motor Drive function blocks are defined in Section 1.2. Please refer there for detailed information about how all these setpoints are used.

Screen Capture of Default Stepper Drive Controls Setpoints

By default, all control sources are selected to be a *'Received CAN J1939 Message'*, and the number is mapped to the appropriate default. See Section 1.8 for more information about the CAN messages, and Table 26 for a complete list of the available control sources supported by the stepper drive.

4.3. PID Control Function Setpoints

This function block is only enabled if the **"Control Source, Motor Mode"** has been set to *'PID Function Block Control.'* Otherwise, all setpoints will read "Not Applicable with this type of Control." Please refer to Section 1.3 for detailed information about how to use these setpoints.
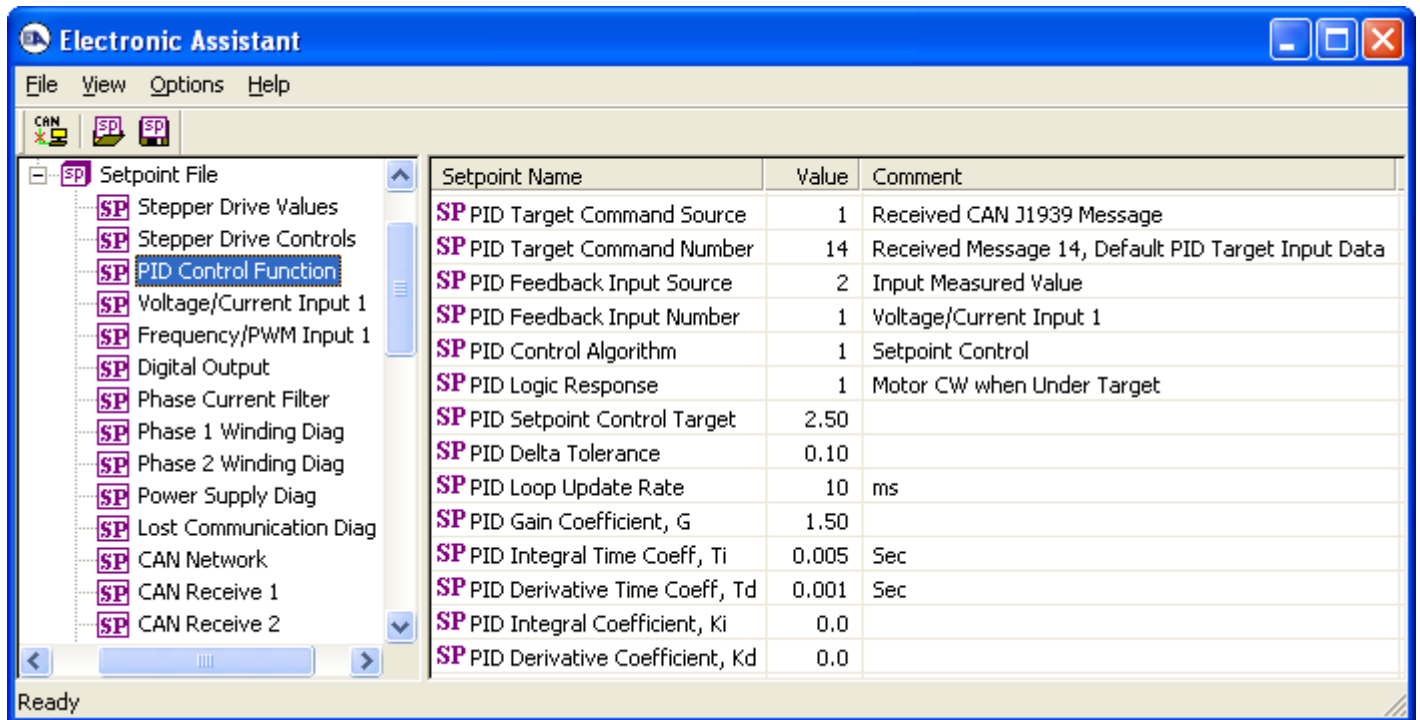
Screen Capture of Default PID Control Function Setpoints

Name	Range	Default	Notes
PID Target Command Source	Drop List	1, Received CAN J1939 Message	See Table 26
PID Target Command Number	Per Source	14, Received Message 14	See Table 26
PID Feedback Input Source	Drop List	2, Input Measured Value	See Table 26
PID Feedback Input Number	Per Source	1, Voltage/Current Input 1	See Table 26
PID Control Algorithm	Drop List	1, Setpoint Control	See Table 7
PID Logic Response	Drop List	1, Motor CW when Under Target	See Table 8
PID Setpoint Control Target	-1,000,000 to 1,000,000	2.50 [V]	In Feedback Input unit
PID Delta Tolerance	0 to 100	0.10 [V]	In Feedback Input unit
PID Loop Update Rate	1 to 60,000 ms	10 ms	1 ms resolution
PID Gain Coefficient, G	0.1 to 10	1.5	For proportional response
PID Integral Time Coeff, Ti	0.001 to 10 Sec	0.005 Sec	0.001 Sec (1ms) resolution
PID Derivative Time Coeff, Td	0.001 to 10 Sec	0.001 Sec	0.001 Sec (1ms) resolution
PID Integral Coefficient, Ki	0 to 10	0.0	0 disables integral, PD ctrl
PID Derivative Coefficient, Kd	0 to 10	0.0	0 disables derivative, PI ctrl

4.4. Voltage/Current Input Setpoints

The Input function blocks are defined in Section 1.4. Please refer there for detailed information about how all these setpoints are used.
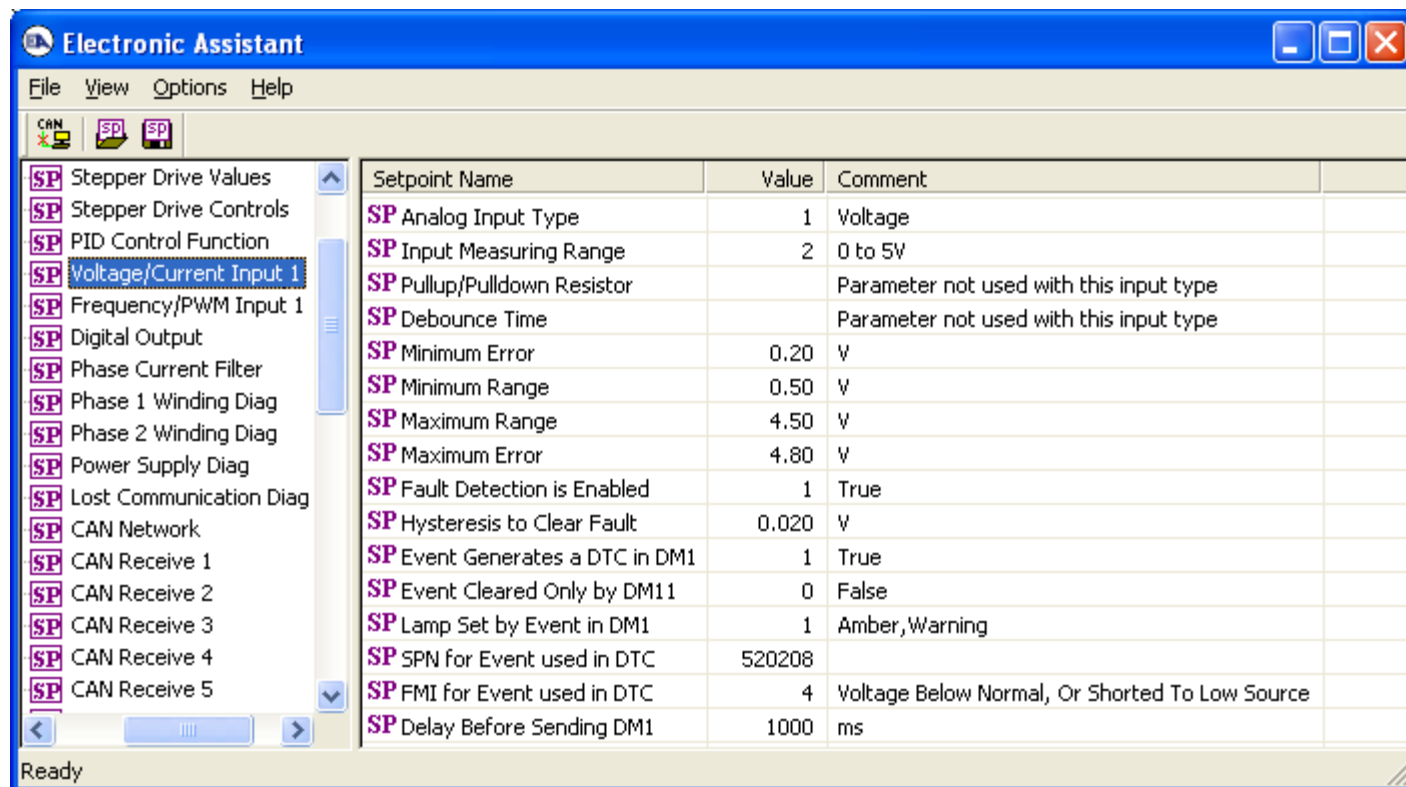
Screen Capture of Default Voltage/Current Input 1 Setpoints

Name	Range	Default	Notes
Analog Input Type	Drop List	1, Voltage	See Table 10
Input Measuring Range	Per Type	2, 0 to 5V	See Tables 13 and 14
Pullup/Pulldown Resistor	Drop List	0, Pullup/down Off	See Table 12
Debounce Time	Drop List	2, 1.78us	See Table 17, Only used with "Digital" input type
Minimum Error	0.0 to Minimum Range	Per Type and Range	Default values for these setpoints are dependent on Type and Range. These values are automatically updated by the ECU when the Type/Range setpoints are changed. See Table 28 for the complete list.
Minimum Range	Minimum Error to Maximum Range	Per Type and Range	
Maximum Range	Minimum Range to Maximum Error	Per Type and Range	
Maximum Error	Maximum Range to Limit (see notes)	Per Type and Range	
Fault Detection is Enabled	False or True	True	See Section 1.7
Hysteresis to Clear Fault	0.01 to 50.00	Per Type and Range	See Table 28
Event Generates a DTC in DM1	False or True	True	See Section 1.7
Event Cleared Only by DM11	False or True	False	See Section 1.7
Lamp Set by Event in DM1	Drop List	1, Amber, Warning	See Section 1.7
SPN for Event used in DTC	1 to 524287	520208 ($7F010)	See Section 1.7 and 3.4
FMI for Event used in DTC	Drop List	Per Type	See Section 1.7 (low FMI)
Delay Before Sending DM1	0 to 60,000 ms	1000 ms	See Section 1.7

Type	Range	Minimum Error	Minimum Range	Maximum Range	Maximum Error*	Hysteresis
Voltage	0-2.5V	0.10 [V]	0.25 [V]	2.50 [V]	2.75 [V]	0.050 [V]
	0-5V	**0.20 [V]**	**0.50 [V]**	**4.50 [V]**	**4.80 [V]**	**0.100 [V]**
	0-10V	0.20 [V]	0.50 [V]	9.50 [V]	9.80 [V]	0.250 [V]
Current	0-20mA	0.00 [mA]	0.00 [mA]	20.00 [mA]	22.00 [mA]	0.250 [mA]
	4-20mA	1.00 [mA]	4.00 [mA]	20.00 [mA]	22.00 [mA]	0.250 [mA]
Frequency or **RPM**	RPM	50 [RPM]	100 [RPM]	2500 [RPM]	2600 [RPM]	10 [RPM]
	0.5 to 50 Hz	**0.25 [Hz]**	**1.00 [Hz]**	**50.00 [Hz]**	**55.00 [Hz]**	**0.200 [Hz]**
	10Hz to 1kHz	7.5 [Hz]	10.0 [Hz]	1000.0 [Hz]	1010.0 [Hz]	5.0 [Hz]
	100Hz to 10kHz	75 [Hz]	100 [Hz]	10000 [Hz]	10100 [Hz]	10 [Hz]
PWM	Low Freq [<1kHz]	1.00 [%]	5.00 [%]	95.00 [%]	99.00 [%]	1.00 [%]
	High Freq [>100Hz]	1.00 [%]	5.00 [%]	95.00 [%]	99.00 [%]	1.00 [%]
Digital	N/A	N/A	0.00	1.00	N/A	N/A

*The limit for the **"Maximum Error"** setpoint will be the upper end of the Range + 10%*

Table 28 – Input Setpoint Defaults Based on Type and Range

4.5. Frequency/PWM Input Setpoints

The Input function blocks are defined in Section 1.4. Please refer there for detailed information about how all these setpoints are used.

Screen Capture of Frequency/PWM Input 1 Setpoints

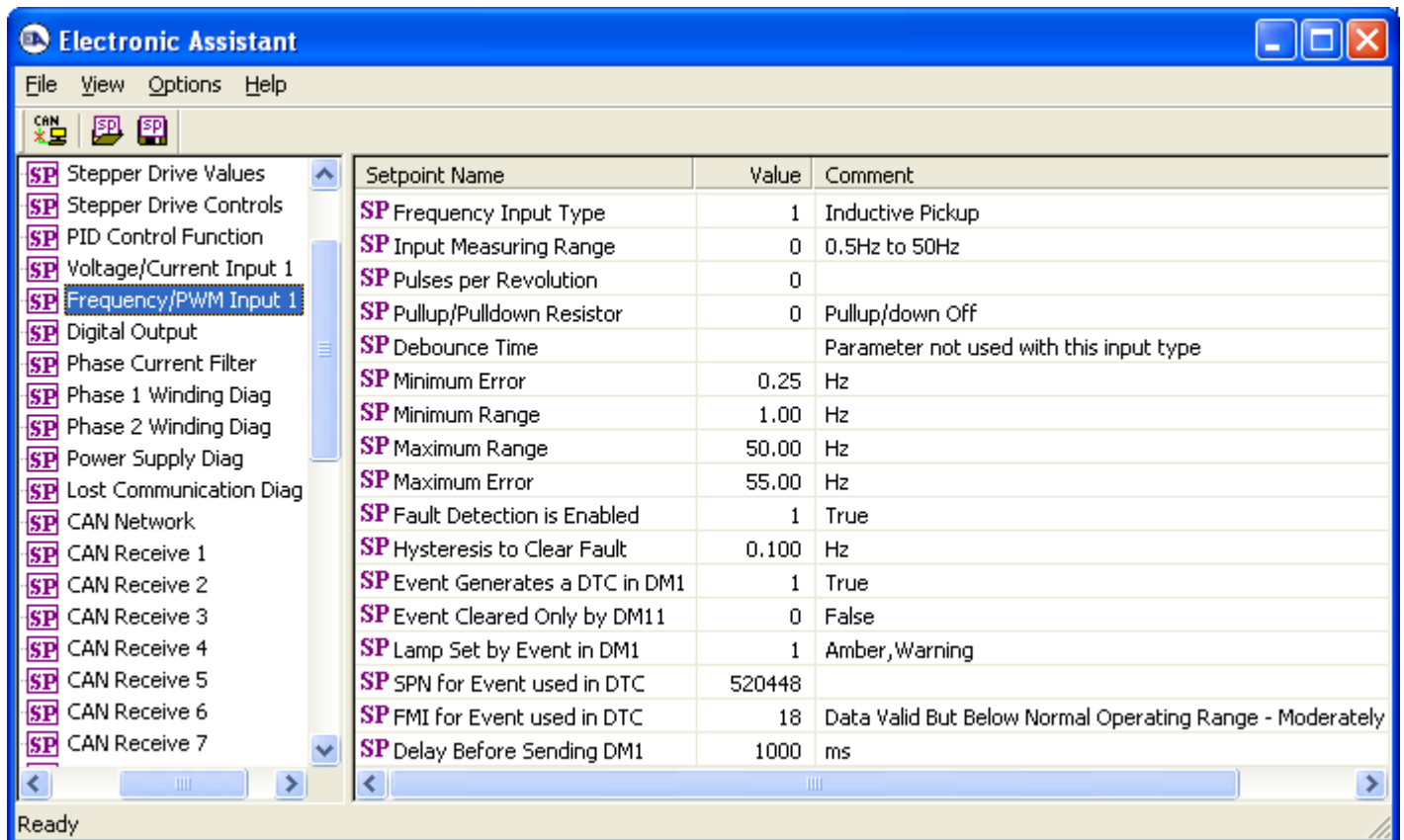| Name	Range	Default	Notes
Frequency Input Type	Drop List	1, Inductive Pickup	See Table 11
Input Measuring Range	Per Type	0, 0.5Hz to 50Hz	See Tables 15 and 16
Pulse per Revolution	0 to 1000	0	Data measured in Hertz
Pullup/Pulldown Resistor	Drop List	0, Pullup/down Off	See Table 12
Debounce Time	Drop List	2, 1.78us	See Table 17, Only used with "Digital" input type
Minimum Error	0 to Minimum Range	Per Type and Range	Default values for these setpoints are dependent on Type and Range. These values are automatically updated by the ECU when the Type/Range setpoints are changed. See Table 28 for the complete list.
Minimum Range	Minimum Error to Maximum Range	Per Type and Range	
Maximum Range	Minimum Range to Maximum Error	Per Type and Range	
Maximum Error	Maximum Range to Limit (see notes)	Per Type and Range	
Fault Detection is Enabled	False or True	True	See Section 1.7
Hysteresis to Clear Fault	0.01 to 50.00	Per Type and Range	See Table 28
Event Generates a DTC in DM1	False or True	True	See Section 1.7
Event Cleared Only by DM11	False or True	False	See Section 1.7
Lamp Set by Event in DM1	Drop List	1, Amber, Warning	See Section 1.7
SPN for Event used in DTC	1 to 524287	520448 ($7F100)	See Section 1.7 and 3.4
FMI for Event used in DTC	Drop List	Per Type	See Section 1.7 (low FMI)
Delay Before Sending DM1	0 to 60,000 ms	1000 ms	See Section 1.7

4.6. Digital Output Setpoints

The Digital Output function block is defined in Section 1.5. Please refer there for detailed information about how all these setpoints are used.
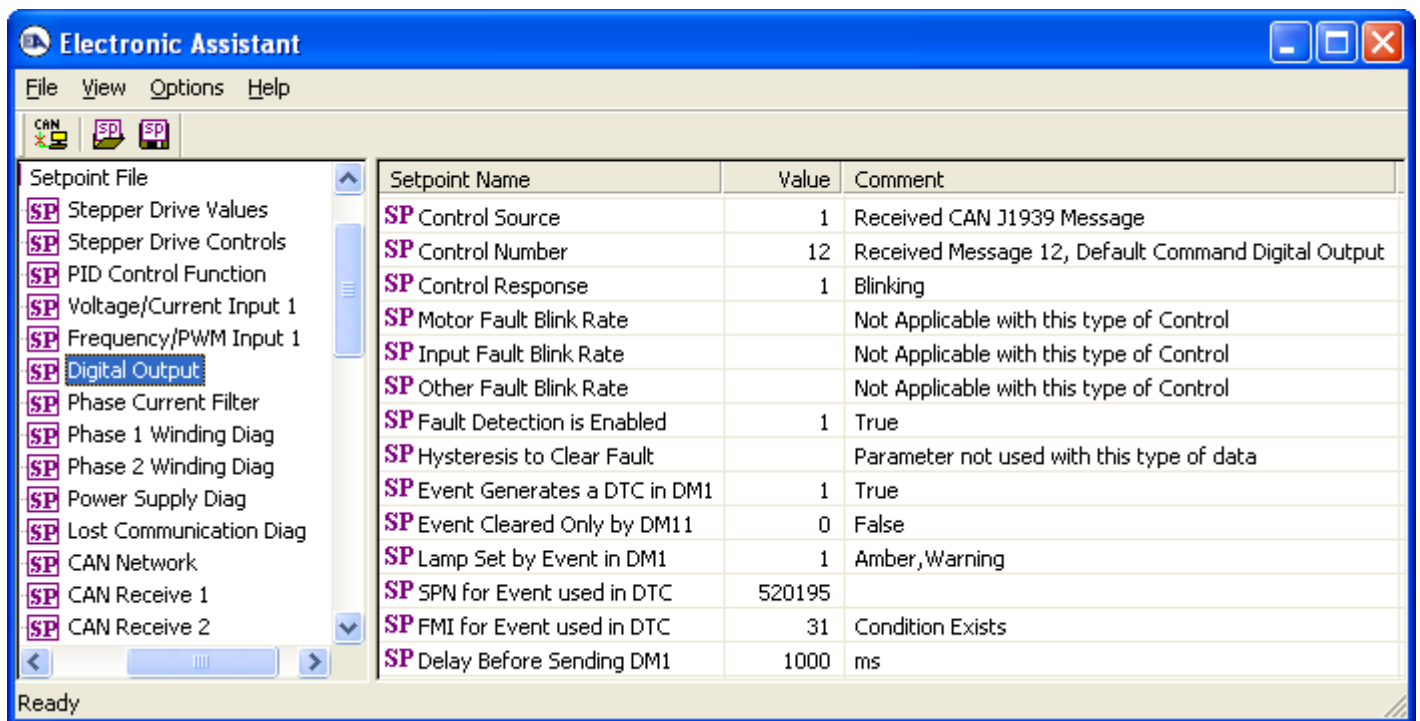

Screen Capture of Default Digital Output Setpoints

Name	Range	Default	Notes
Control Source	Drop List	1, Received CAN J1939 Message	See Table 26
Control Number	Per Source	12, Received Message 12	See Table 26
Control Response	Drop List	1, Blinking	See Table 19
Motor Fault Blink Rate	100 to 5,000 ms	100 ms	Default blink rate
Input Fault Blink Rate	100 to 5,000 ms	500 ms	
Other Fault Blink Rate	100 to 5,000 ms	2000 ms	
Fault Detection is Enabled	False or True	True	See Section 1.7
Hysteresis to Clear Fault	N/A	N/A	
Event Generates a DTC in DM1	False or True	True	See Section 1.7
Event Cleared Only by DM11	False or True	False	See Section 1.7
Lamp Set by Event in DM1	Drop List	1, Amber, Warning	See Section 1.7
SPN for Event used in DTC	1 to 524287	520195 ($7F003)	See Section 1.7 and 3.4
FMI for Event used in DTC	Drop List	31, Condition Exists	See Section 1.7
Delay Before Sending DM1	0 to 60,000 ms	1000 ms	See Section 1.7

4.7. Phase Current Filter Setpoints

The Phase Current Filter function block is defined in Section 1.6. Please refer there for detailed information about how all these setpoints are used. The same type of filter is applied to both phases.
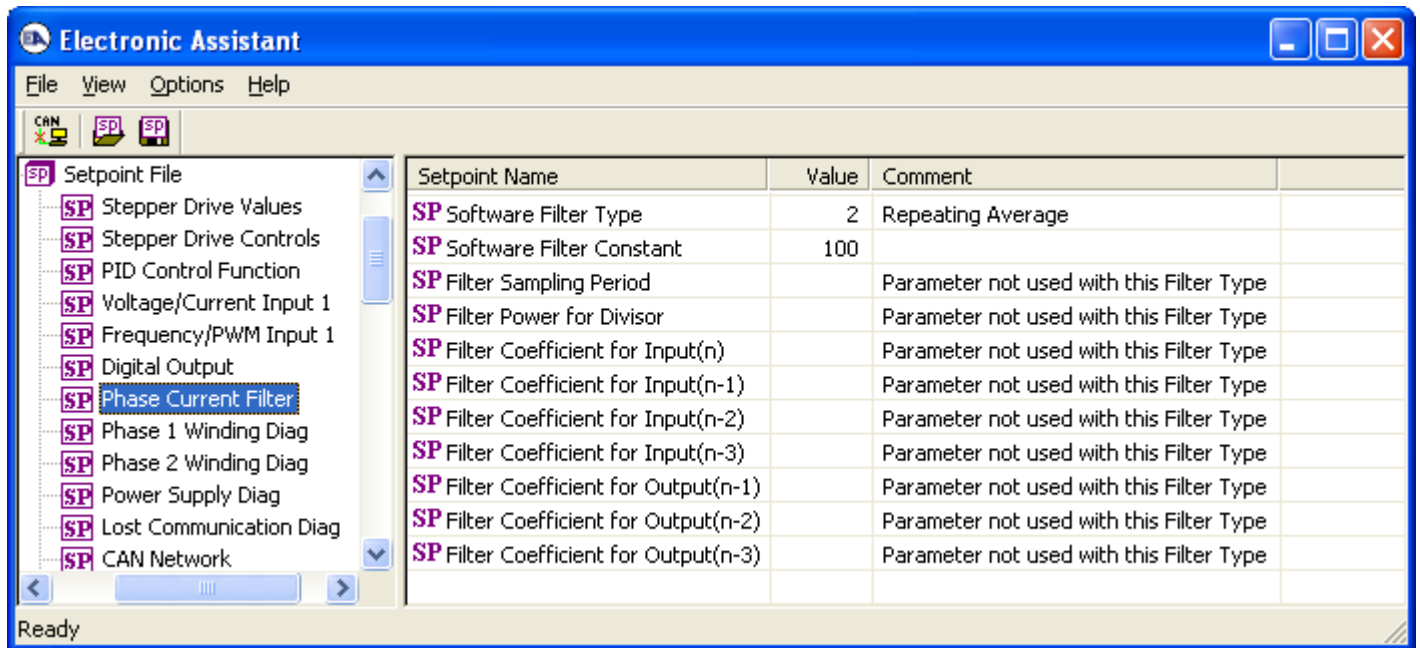

Software Filter Type	Drop List	2, Repeating Average	See Table 20
Software Filter Constant	1 to 1,000	100	Used with Moving or Repeating Average Filters
Filter Sampling Period	1 to 1,000 ms	10 ms	Used with 3rd Order Low Pass, otherwise called every 1ms
Filter Power for Divisor	0 to 16	10	2^Power is used in fixed point math for 3rd Order Low Pass
Filter Coefficient for Input(n)	-10,000 to 10,000	120	Used with 3rd Order Low Pass X = Y*2^Power
Filter Coefficient for Input(n-1)	-10,000 to 10,000	241	See above
Filter Coefficient for Input(n-2)	-10,000 to 10,000	120	See above
Filter Coefficient for Input(n-3)	-10,000 to 10,000	0	See above
Filter Coefficient for Output(n-1)	-10,000 to 10,000	704	See above
Filter Coefficient for Output(n-2)	-10,000 to 10,000	-164	See above
Filter Coefficient for Output(n-3)	-10,000 to 10,000	0	See above

4.8. [Additional] Diagnostic Setpoints

Here, the [Additional] diagnostics are Phase 1 Winding, Phase 2 Winding, Power Supply, and Lost Communication

The Diagnostic function block is defined in Section 1.7. Please refer there for detailed information about how all these setpoints are used.
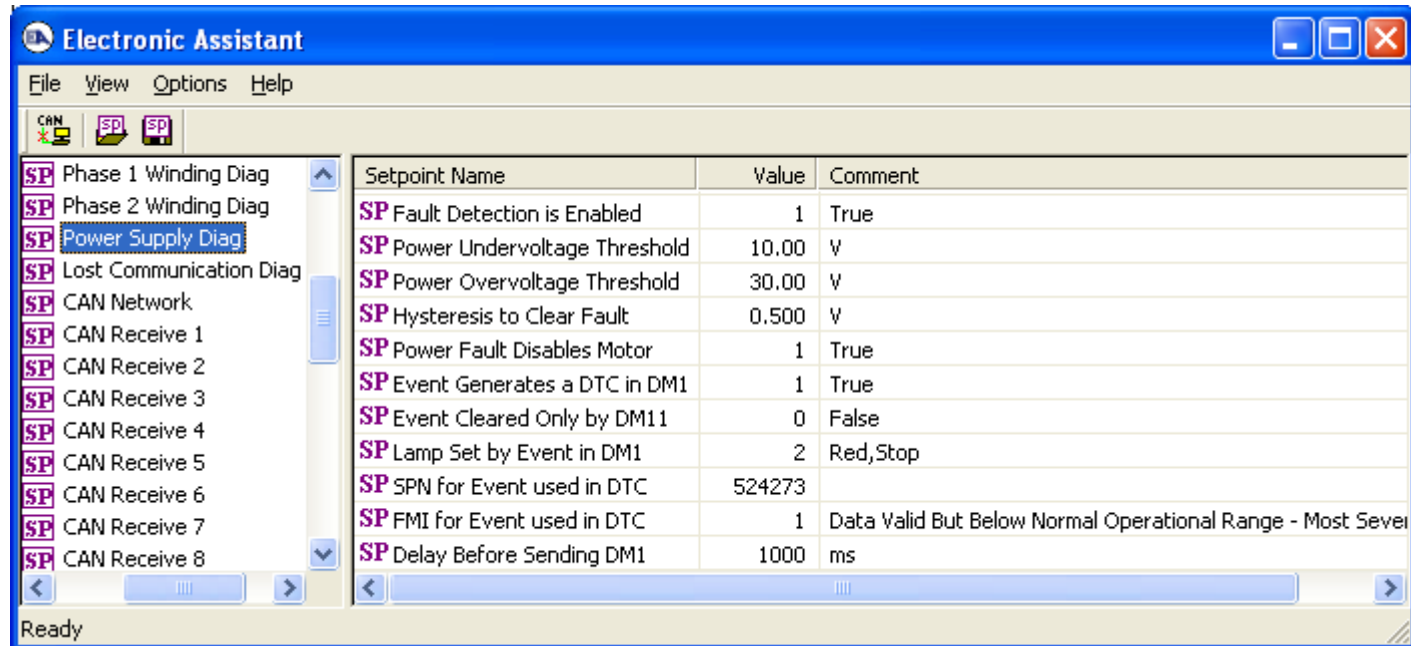
Screen Capture of Default Power Supply Diagnostic Setpoints

Name	Range	Default		Notes
Fault Detection is Enabled	False or True	True		See Section 1.7
Power Undervoltage Threshold *	9 to Over V	10.00 V		See * below
Power Overvoltage Threshold *	Under to 32V	30.00 V		See * below
Phase Current Fault Threshold **	5 to 500mA	50.00 mA		See ** below
Hysteresis to Clear Fault	0.01 to 50.0	10.000 mA 0.500 V N/A	Current Power Comm	See Section 1.7
Power Fault Disables Motor *	False or True	True		See * below
Event Generates a DTC in DM1	False or True	True (False for Lost Communication)		See Section 1.7
Event Cleared Only by DM11	False or True	False		See Section 1.7
Lamp Set by Event in DM1	Drop List	1, Amber, Warning 2, Red, Stop 1, Amber, Warning	Current Power Comm	See Section 1.7
SPN for Event used in DTC	1 to 524287	520193 ($7F001) 520194 ($7F002) 524273 ($7FFF1) 524287 ($7FFFF)	Phase1 Phase2 Power Comm	See Section 1.7 and 3.4
FMI for Event used in DTC	Drop List	5, Current Below Normal, Open 1, Data Below Normal - Sever 19, Network Error	Current Power Comm	See Section 1.7
Delay Before Sending DM1	0 to 60,000 ms	1000 ms		See Section 1.7

* Only used with Power Supply Diag group
** Only used with Phase X Winding Diag group, where X = 1 or 2

4.9. CAN Network Setpoints

The CAN Network setpoints are described in detail in Section 3.2. Also, refer to the notes for more information about each setpoint.

Screen Capture of Default CAN Network Setpoints

Name	Range	Default	Notes
ECU Instance Number	Drop List	0, #1 – First Instance	Per J1939-81
ECU Address	0 to 253	200	Preferred address for a motor control per J1939
AtoD Filter Frequency	Drop List	2, 60Hz	0 = None, 1 = 50Hz, 2 = 60Hz, 3 = Both This filter used on all values measured by ADC

4.10. CAN Receive Setpoints

The CAN Receive function block is defined in Section 1.8, with addition information in Section 3.3. Please refer there for detailed information about how all these setpoints are used.
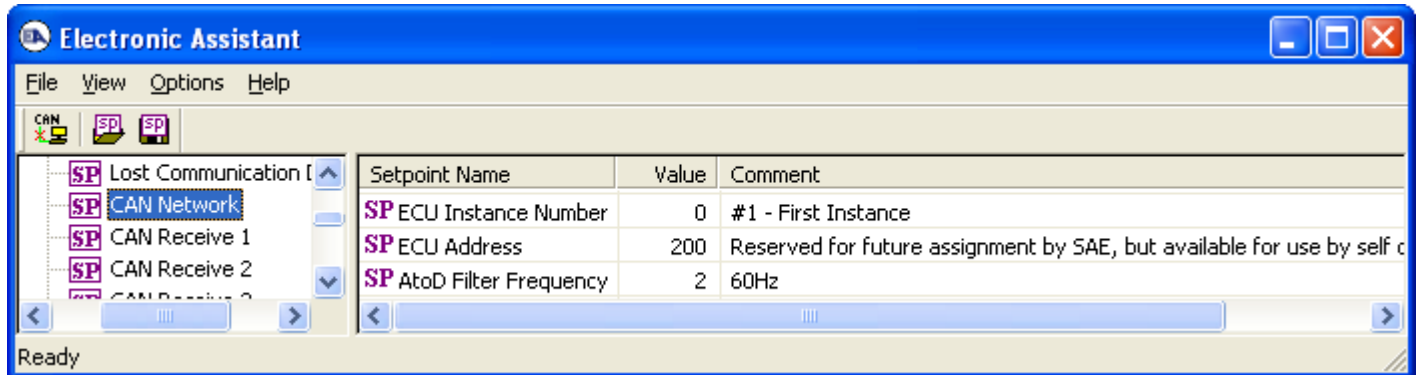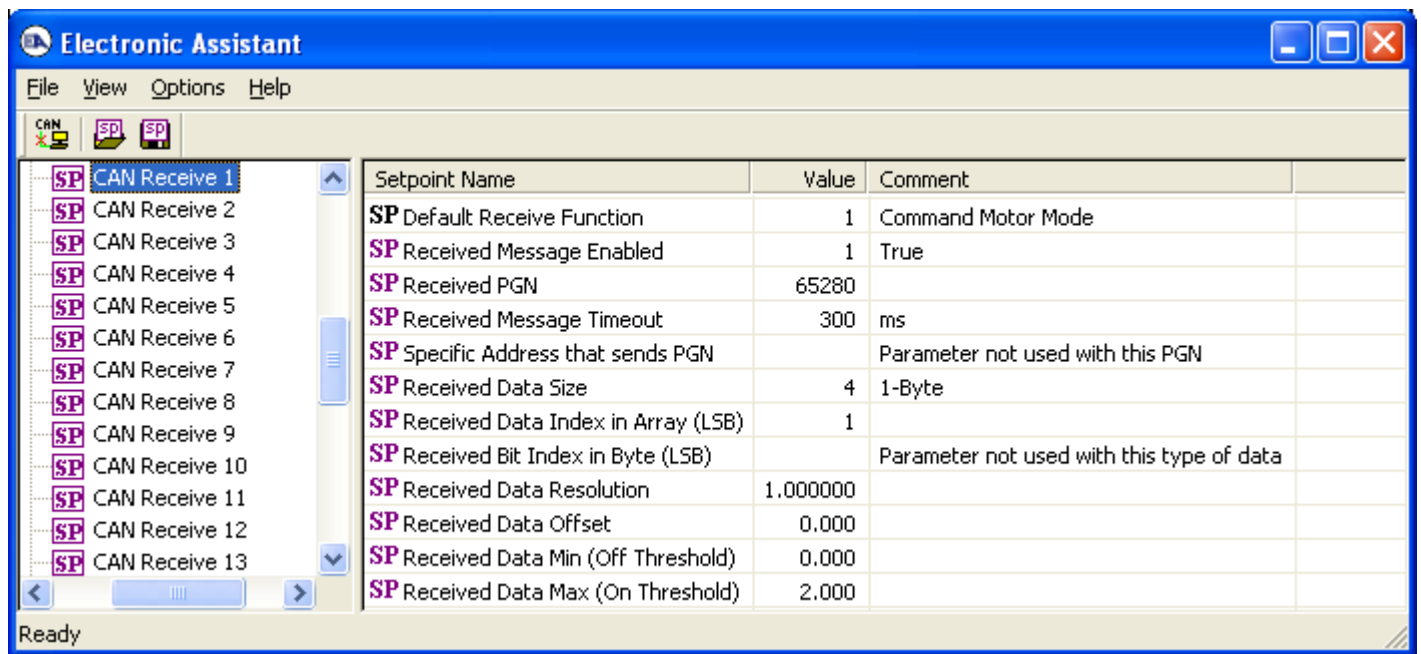

Screen Capture of Default CAN Receive 1 Setpoints

Name	Range	Default	Notes
Default Receive Function	N/A	Different for each	Read-Only Setpoint See Table 23 for defaults
Received Message Enabled	False or True	True	
Received PGN	0 to 65535	Different for each	See Section 3.3 for defaults
Received Message Timeout	0 to 60,000 ms	300 ms	Expects all data at 100ms
Specific Address that sends PGN	0 to 255	254 (0xFE, Null Address)	Not used by default
Receive Data Size	Drop List	Different for each	0 = Not Used (disabled) 1 = 1-Bit 2 = 2-Bits 3 = 4-Bits 4 = 1-Byte 5 = 2-Bytes 6 = 4-Bytes See Section 3.3 for defaults
Receive Data Index in Array (LSB)	1 to 8-DataSize	Different for each	See Section 3.3 for defaults
Receive Bit Index in Byte (LSB)	1 to 8-BitSize	Different for each	Only used with Bit Data Types
Receive Data Resolution	-1,000,000 to 1,000,000	Different for each	See Section 3.3 for defaults
Receive Data Offset	-10,000 to 10,000	Different for each	See Section 3.3 for defaults
Receive Data Min (Off Threshold)	-1,000,000 to Rxd Data Max	Different for each	See Section 3.3 for defaults
Receive Data Max (On Threshold)	Rxd Data Min to 1,000,000	Different for each	See Section 3.3 for defaults

4.11. CAN Transmit Setpoints

The CAN Transmit function block is defined in Section 1.9, with addition information in Section 3.4. Please refer there for detailed information about how all these setpoints are used.
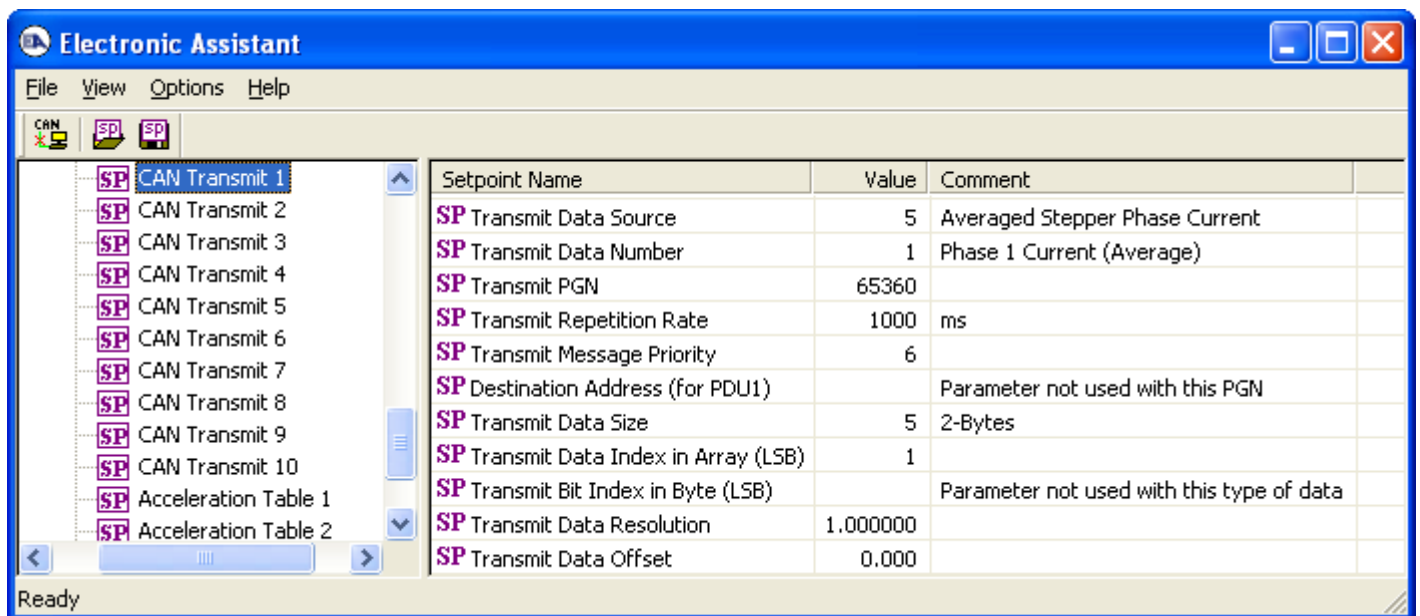

Screen Capture of Default CAN Transmit 1 Setpoints

Name	Range	Default	Notes
Transmit Data Source	Drop List	Different for each	See Table 24 for defaults
Transmit Data Number	Per Source	Different for each	See Table 26 for options
Transmit PGN	0 to 65535	Different for each	See Section 3.4 for defaults
Transmit Repetition Rate	0 to 60,000 ms	1000 ms (or 0ms)	Only on the first Txd block using the PGN, otherwise 0
Transmit Message Priority	0 to 7	6	Proprietary B Priority
Destination Address (for PDU1)	0 to 255	254 (0xFE, Null Address)	Not used by default
Transmit Data Size	Drop List	Different for each	0 = Not Used (disabled) 1 = 1-Bit 2 = 2-Bits 3 = 4-Bits 4 = 1-Byte 5 = 2-Bytes 6 = 4-Bytes See Section 3.4 for defaults
Transmit Data Index in Array (LSB)	1 to 8-DataSize	Different for each	See Section 3.4 for defaults
Transmit Bit Index in Byte (LSB)	1 to 8-BitSize	Different for each	Only used with Bit Data Types
Transmit Data Resolution	-1,000,000 to 1,000,000	Different for each	See Section 3.4 for defaults
Transmit Data Offset	-10,000 to 10,000	Different for each	See Section 3.4 for defaults

4.12. Acceleration Table Setpoints

See Section 1.2 for information on how to use the Acceleration Tables

| Tree	Setpoint Name	Value	Comment	
J1939 CAN Network	Point 1 Step Rate	15	Step/s	
ECU AX100700, 2A Stepper Motor Drive #1	Point 2 Step Rate	25	Step/s	
General ECU Information	Point 3 Step Rate	33	Step/s	
Setpoint File	Point 4 Step Rate	40	Step/s	
Stepper Drive Values	Point 5 Step Rate	46	Step/s	
Stepper Drive Controls	Point 6 Step Rate	51	Step/s	
PID Control Function	Point 7 Step Rate	55	Step/s	
Voltage/Current Input 1	Point 8 Step Rate	58	Step/s	
Frequency/PWM Input 1	Point 9 Step Rate	60	Step/s	
Digital Output	Point 10 Step Rate	62	Step/s	
Phase Current Filter	Point 11 Step Rate	63	Step/s	
Phase 1 Winding Diag	Point 12 Step Rate	64	Step/s	
Phase 2 Winding Diag	Point 13 Step Rate	65	Step/s	
Power Supply Diag	Point 14 Step Rate	65	Step/s	
Lost Communication Diag	Point 15 Step Rate	65	Step/s	
CAN Network	Point 16 Step Rate	65	Step/s	
CAN Receive 1	Point 17 Step Rate	65	Step/s	
CAN Receive 2	Point 18 Step Rate	65	Step/s	
CAN Receive 3	Point 19 Step Rate	65	Step/s	
CAN Receive 4	Point 20 Step Rate	65	Step/s	
CAN Receive 5	Point 21 Step Rate	65	Step/s	
CAN Receive 6	Point 22 Step Rate	65	Step/s	
CAN Receive 7	Point 23 Step Rate	65	Step/s	
CAN Receive 8	Point 24 Step Rate	65	Step/s	
CAN Receive 9	Point 25 Step Rate	65	Step/s	
CAN Receive 10	Point 1 Time	10	ms	
CAN Receive 11	Point 2 Time	20	ms	
CAN Receive 12	Point 3 Time	30	ms	
CAN Receive 13	Point 4 Time	40	ms	
CAN Receive 14	Point 5 Time	50	ms	
CAN Receive 15	Point 6 Time	60	ms	
CAN Transmit 1	Point 7 Time	70	ms	
CAN Transmit 2	Point 8 Time	80	ms	
CAN Transmit 3	Point 9 Time	90	ms	
CAN Transmit 4	Point 10 Time	100	ms	
CAN Transmit 5	Point 11 Time	110	ms	
CAN Transmit 6	Point 12 Time	120	ms	
CAN Transmit 7	Point 13 Time	130	ms	
CAN Transmit 8	Point 14 Time	140	ms	
CAN Transmit 9	Point 15 Time	150	ms	
CAN Transmit 10	Point 16 Time	160	ms	
Acceleration Table 1	Point 17 Time	170	ms	
Acceleration Table 2	Point 18 Time	180	ms	
Acceleration Table 3	Point 19 Time	190	ms	
Acceleration Table 4	Point 20 Time	200	ms	
Acceleration Table 5	Point 21 Time	210	ms	
	Point 22 Time	220	ms	
	Point 23 Time	230	ms	
	Point 24 Time	240	ms	
	Point 25 Time	250	ms	

Screen Capture of Default Acceleration Table Setpoints

5. USING RS-232 WITH TERA TERM

Additional information for diagnostics or testing is available through RS-232. using the RS-232 connection available on the board at header J3. Use the RS-232 "Programming Interface" and cable provide by Axiomatic. Note: The TE Deutsch box will have to be opened to access header J3.

- Connect the 5-Pin header cable to J3. Connect the DB-9 on the interface board to a COM port
  Set the switch SW2 "Power" to the side that reads PC
  Set the switch SW1 "Boot" to the side that reads ON

- Open Tera Term Pro, and set it up as shown in the steps below.
  (Free downloadable from Hhttp://hp.vector.co.jp/authors/VA002416/teraterm.htmlH)

- Select **Serial** with the appropriate COM port for your PC or laptop
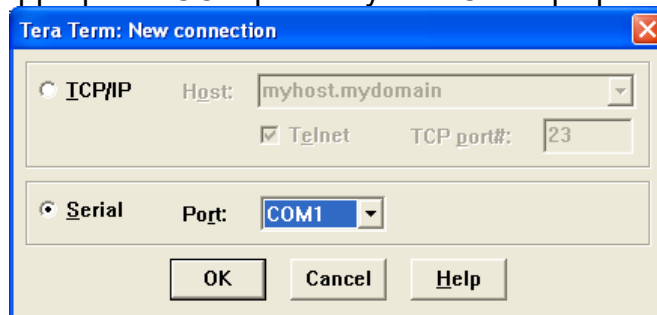
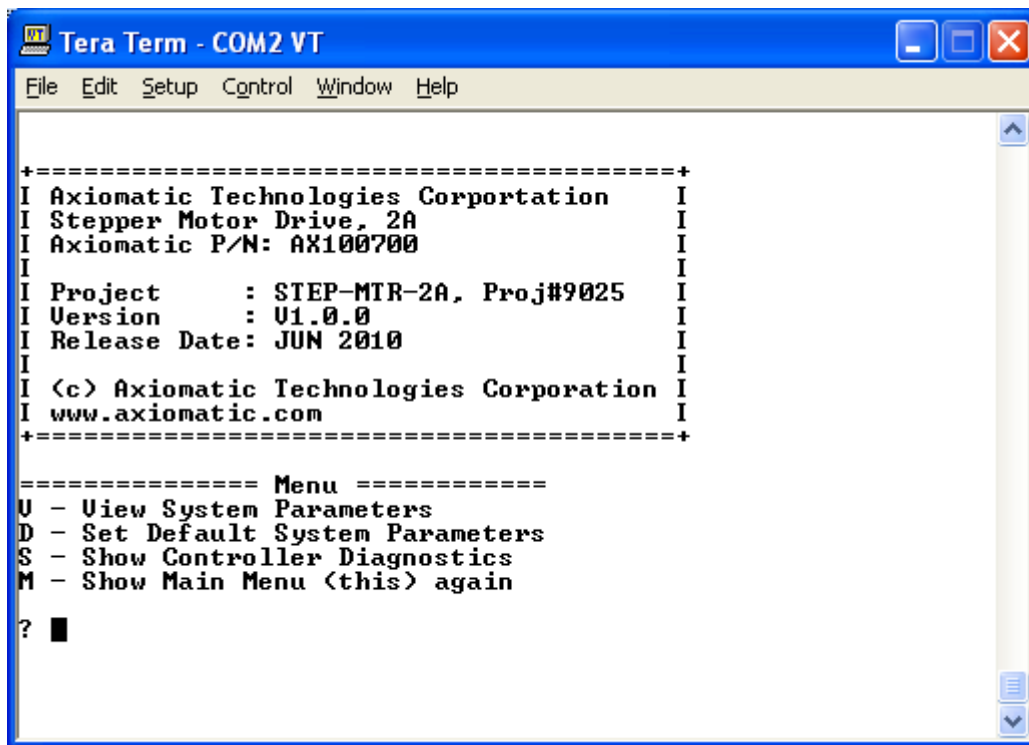- Go to **Setup/Serial Port** and change the settings to exactly as shown below (other than Port)
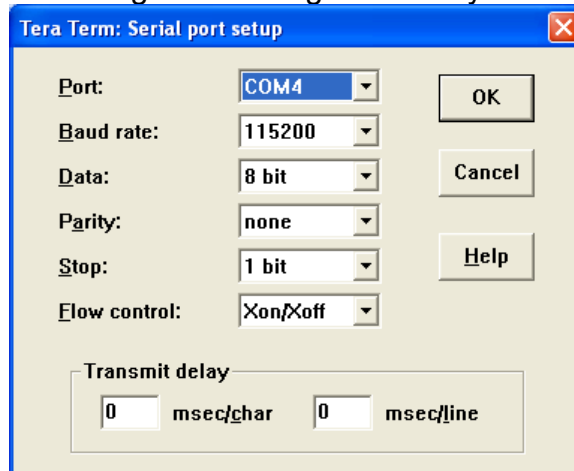
- Go to **Setup/Terminal** and verify that New-line Transmit and Receive are CR. The window size can be adjusted as desired by checking 'Term size = win size'
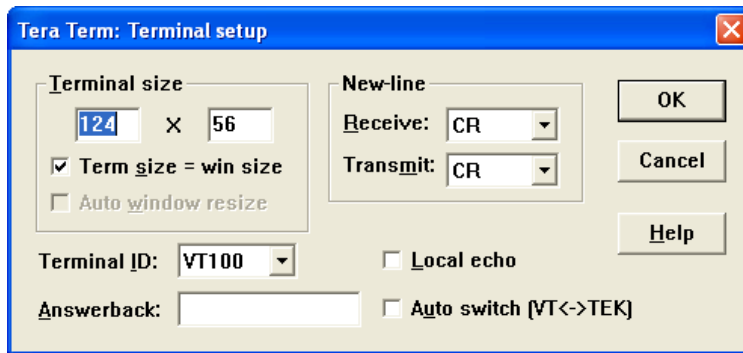
5.1. Main Menu Options

Apply power to the Stepper Drive, and the power up banner will be displayed.


```
+==========================================+
I Axiomatic Technologies Corportation    I
I Stepper Motor Drive, 2A                 I
I Axiomatic P/N: AX100700                 I
I                                         I
I Project      : STEP-MTR-2A, Proj#9025   I
I Version      : V1.0.0                    I
I Release Date: JUN 2010                  I
I                                         I
I (c) Axiomatic Technologies Corporation I
I www.axiomatic.com                       I
+==========================================+

============== Menu ============
V - View System Parameters
D - Set Default System Parameters
S - Show Controller Diagnostics
M - Show Main Menu (this) again

? █
```

V – View System Parameters

The "View Parameters" option **'v'** allows the user to see the same information as available through the Axiomatic Electronic Assistant. However, changing the setpoints is not possible via RS-232, so the CAN link must be used.

The information about the different function blocks will be dumped to the screen in readable sections. To move to the next section in the list, hit any key. To exit the data dump before reaching the end of the setpoint list, hit '**x**' to exit at any time. While in 'viewing' mode, the motor should be disabled, as no CAN messages will be processed.

```
Tera Term - COM2 VT
File  Edit  Setup  Control  Window  Help

v


Stepper Defaults
============================
MotorAnglePerStep = 18
StepRateUnits     = St/s
PutDriveToSleep   = False
ResetMotor        = False
EStopMotor        = False
MotorDriveMode    = Stopped
DriveStepType     = SixteenStep
RotationDirection = ClockWise
SelectedAccTable  = NoProfile
PercentFastDecay  = Mixed

PhaseCurrent      = 500 [mA]
NumberOfSteps     = Continuous
StepRate          = 25 [St/s]

Stepper Control
============================
SaveOperatingData:  Control_Source  = ReceivedCANMsg    Control_Number  = 13, DefaultSaveData
PutDriveToSleep  :  Control_Source  = ReceivedCANMsg    Control_Number  =  9, DefaultPutToSleep
ResetMotor       :  Control_Source  = ReceivedCANMsg    Control_Number  = 10, DefaultResetMotor
EStopMotor       :  Control_Source  = ReceivedCANMsg    Control_Number  = 11, DefaultEStopMotor
MotorDriveMode   :  Control_Source  = PIDMotorControl   Control_Number  =  2
DriveStepType    :  Control_Source  = ReceivedCANMsg    Control_Number  =  6, DefaultStepType
RotationDirection:  Control_Source  = PIDMotorControl   Control_Number  =  3
SelectedAccTable :  Control_Source  = ControlNotUsed    Control_Number  =  8
PercentFastDecay :  Control_Source  = ReceivedCANMsg    Control_Number  =  7, DefaultPFD
PhaseCurrent     :  Control_Source  = ReceivedCANMsg    Control_Number  =  4, DefaultPhaseCurr
NumberOfSteps    :  Control_Source  = ControlNotUsed    Control_Number  =  5
StepRate         :  Control_Source  = PIDMotorControl   Control_Number  =  4

Stepper Limits
============================
PhaseCurrent     :  Minimum_Range  = 100.000   [mA]     Maximum_Range  = 2000.000 [mA]
NumberOfSteps    :  Minimum_Range  = 0.000     [mA]     Maximum_Range  = 1000.000 [mA]
StepRate         :  Minimum_Range  = 1.000     [St/s]   Maximum_Range  = 50.000   [St/s]
```

```
Tera Term - COM2 VT

File  Edit  Setup  Control  Window  Help

Analog Input (Ain1/Din1)
===========================
InputType          = Voltage
InputRange         = 0 to 5V
PullupOrPulldown   = Pullup/down Off
DebounceFilter     = 1.78us
NumPulsePerRev     = 0

InputErrorMinimum  = 0.20     [V]
InputRangeMinimum  = 0.50     [V]
InputRangeMaximum  = 4.50     [V]
InputErrorMaximum  = 4.80     [V]

Freq Input (Fin1/Din2)
==============================
InputType          = Inductive
InputRange         = 0.5Hz to 50Hz
PullupOrPulldown   = Pullup/down Off
DebounceFilter     = 1.78us
NumPulsePerRev     = 0

InputErrorMinimum  = 0.25     [Hz]
InputRangeMinimum  = 1.00     [Hz]
InputRangeMaximum  = 50.00    [Hz]
InputErrorMaximum  = 55.00    [Hz]
```

```
Tera Term - COM2 VT

File  Edit  Setup  Control  Window  Help

Digital Output
===========================
Control_Source     = ReceivedCANMsg      Control_Number    = 12, DefaultDoutCtrl
ControlResponse    = Blinking
BlinkRate_Fault_1  = 100
BlinkRate_Fault_2  = 500
BlinkRate_Fault_3  = 2000

Extra Diagnostics
===========================
ADCFilterFreq      = 60Hz
PhaseCurrErrThres  = 50.00    [mA]
SetUnderVoltage    = 10.00    [V]
SetOverVoltage     = 30.00    [V]
UpsFaultShutdown   = True

Phase Current Filter
===========================
DataFilterType     = Repeating Average
DataFilterConst    = 100

Closed-Loop Motor Control
===========================
PIDCtrlResponse    = FixedSetpointControl
SetpointValue      = 2.50
FeedbackInput      = Control_Source    = MeasuredInput      Control_Number    = 1, AnalogInput
PID_LoopCallRate   = 10 [ms]
PID_Gain           = 1.500
PID_Ti             = 0.00500
PID_Td             = 0.00100
PID_Ki             = 0.00000
PID_Kd             = 0.00000    x

+==========================================+
I Axiomatic Technologies Corportation    I
I Stepper Motor Drive, 2A                 I
I Axiomatic P/N: AX100700                 I
I                                         I
I Project     : STEP-MTR-2A, Proj#9025    I
I Version     : V1.0.0                     I
I Release Date: JUN 2010                  I
I                                         I
I (c) Axiomatic Technologies Corporation I
I www.axiomatic.com                       I
+==========================================+

============== Menu ============
V - View System Parameters
D - Set Default System Parameters
```
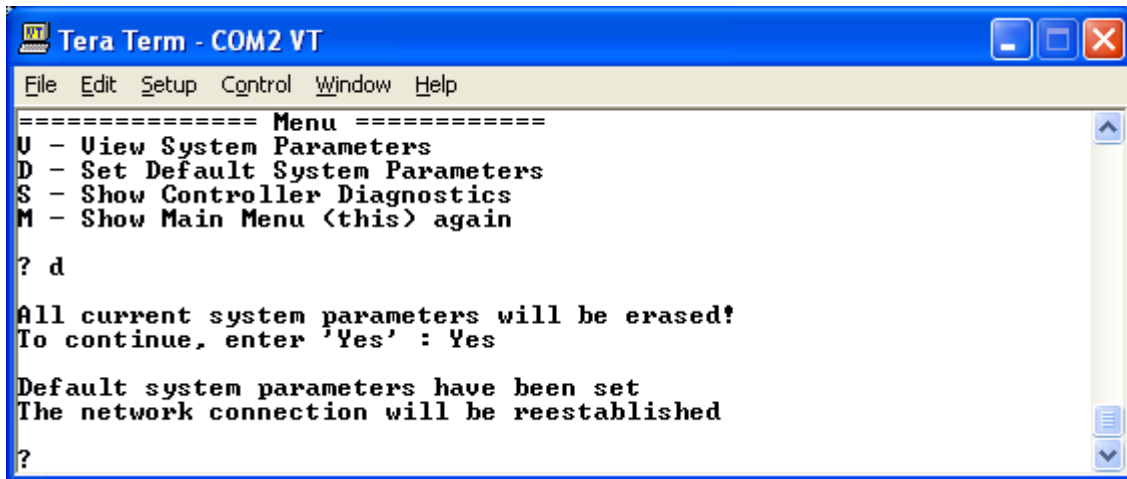
D – Set Default System Parameters

The "Default Parameters" option'd' resets all setpoints to the factory defaults as described in Section 4. The unit will prompt for confirmation that this action should proceed, and enter **'Yes'** to continue. (Note: password is case sensitive)

S – Show Controller Diagnostics

The "Show Diagnostics" option's' will display something like the following screen capture when running normally without errors.

```
Tera Term - COM2 VT

File  Edit  Setup  Control  Window  Help

MEASURED INPUTS
=============================
SupplyVoltageMeas  = 18.83      [V]
Phase1CurrentFB    = 349        [mA]
Phase2CurrentFB    = 314        [mA]
AnalogInputMeas    = 2.33       [V]
FreqInputMeas      = 25.01      [Hz]

OUTPUTS and CONTROLS
=============================
DigitalOutState    = 0, OFF
DoutShortState     = 0, OFF
CtrlOut_UrefValue  = 1230       [mA]

RECEIVED CAN MESSAGES
=============================
DataRxdOnCANMsg1   = 2.00            RxdDefaultMotorMode
DataRxdOnCANMsg2   = 0.00            RxdDefaultDirection
DataRxdOnCANMsg3   = 6.00            RxdDefaultStepRate
DataRxdOnCANMsg4   = 1230.00         RxdDefaultPhaseCurr
DataRxdOnCANMsg5   = 0.00            RxdDefaultNumOfSteps
DataRxdOnCANMsg6   = 3.00            RxdDefaultStepType
DataRxdOnCANMsg7   = 0.00            State = Not Available
DataRxdOnCANMsg8   = 0.00            RxdDefaultAccTable
DataRxdOnCANMsg9   = 0.00            State = Not Available
DataRxdOnCANMsg10  = 0.00            RxdDefaultResetMotor
DataRxdOnCANMsg11  = 0.00            RxdDefaultEStopMotor
DataRxdOnCANMsg12  = 0.00            RxdDefaultDoutCtrl
DataRxdOnCANMsg13  = 0.00            State = Not Available
DataRxdOnCANMsg14  = 0.00            State = Not Available
DataRxdOnCANMsg15  = 0.00            State = Not Available

TRANSMITTED CAN MESSAGES
=============================
DataTxdToCANMsg1   = 349.78          TxdDefaultPhase1Curr
DataTxdToCANMsg2   = 314.39          TxdDefaultPhase2Curr
DataTxdToCANMsg3   = 2.00            TxdDefaultMotorMode
DataTxdToCANMsg4   = 0.00            TxdDefaultDirection
DataTxdToCANMsg5   = 6.00            TxdDefaultStepRate
DataTxdToCANMsg6   = 2.33            TxdDefaultAin1OrDin1
DataTxdToCANMsg7   = 25.01           TxdDefaultFin1OrDin2
DataTxdToCANMsg8   = 0.00            TxdDefaultDoutState
DataTxdToCANMsg9   = 18.83           TxdDefaultSupplyVolt

MOTOR CONTROLS
=============================
StepIn_SaveCtrls   = 0, False
StepIn_GoToSleep   = 0, False
StepIn_ResetMtr    = 0, False
StepIn_EStopMtr    = 0, False

CtrlIn_MotorMode   = 2, Running
CtrlIn_StepType    = 3, SixteenStep
CtrlIn_Direction   = 0, CounterCW
CtrlIn_AccTable    = 0, NoProfile
StepIn_PFDSetting  = 0, Slow

CtrlIn_PhaseCurr   = 1230       [mA]
CtrlIn_NumOfSteps  = 0
CtrlIn_StepRate    = 6.00       [St/s]

CAN Status:
->Address has been claimed successfully
?
```

However, should errors be present, additional information would be available as shown below.

```
MEASURED INPUTS
===========================
SupplyVoltageMeas = 19.08        [V]
Phase1CurrentFB   = 0            [mA]
Phase2CurrentFB   = 0            [mA]
AnalogInputMeas   = 0.00         [V]        State = Data Error       Error = 0X0100, Analog Input Out of Range Low    OC = 2
FreqInputMeas     = 25.01        [Hz]

OUTPUTS and CONTROLS
===========================
DigitalOutState   = 0, OFF
DoutShortState    = 0, OFF
CtrlOut_VrefValue = 1230         [mA]

RECEIVED CAN MESSAGES
===========================
DataRxdOnCANMsg1  = 0.00                    State = Data Error       Error = 0X80000000, CAN Message Receive Timeout
DataRxdOnCANMsg2  = 0.00                    State = Data Error       Error = 0X80000000, CAN Message Receive Timeout
DataRxdOnCANMsg3  = 6.00                    State = Data Error       Error = 0X80000000, CAN Message Receive Timeout
DataRxdOnCANMsg4  = 1230.00                 State = Data Error       Error = 0X80000000, CAN Message Receive Timeout
DataRxdOnCANMsg5  = 0.00                    State = Data Error       Error = 0X80000000, CAN Message Receive Timeout
DataRxdOnCANMsg6  = 3.00                    RxdDefaultStepType
DataRxdOnCANMsg7  = 0.00                    State = Not Available
DataRxdOnCANMsg8  = 0.00                    RxdDefaultAccTable
DataRxdOnCANMsg9  = 0.00                    State = Not Available
DataRxdOnCANMsg10 = 0.00                    RxdDefaultResetMotor
DataRxdOnCANMsg11 = 0.00                    RxdDefaultEStopMotor
DataRxdOnCANMsg12 = 0.00                    RxdDefaultDoutCtrl
DataRxdOnCANMsg13 = 0.00                    State = Not Available
DataRxdOnCANMsg14 = 0.00                    State = Not Available
DataRxdOnCANMsg15 = 0.00                    State = Not Available

TRANSMITTED CAN MESSAGES
===========================
DataTxdToCANMsg1  = 0.69                    TxdDefaultPhase1Curr
DataTxdToCANMsg2  = 0.57                    TxdDefaultPhase2Curr
DataTxdToCANMsg3  = 0.00                    TxdDefaultMotorMode
DataTxdToCANMsg4  = 0.00                    TxdDefaultDirection
DataTxdToCANMsg5  = 6.00                    TxdDefaultStepRate
DataTxdToCANMsg6  = 0.00                    State = Data Error       Error = 0X0100, Analog Input Out of Range Low    OC = 2
DataTxdToCANMsg7  = 25.01                   TxdDefaultFin1OrDin2
DataTxdToCANMsg8  = 0.00                    TxdDefaultDoutState
DataTxdToCANMsg9  = 19.08                   TxdDefaultSupplyVolt

MOTOR CONTROLS
===========================
StepIn_SaveCtrls  = 0, False
StepIn_GoToSleep  = 0, False
StepIn_ResetMtr   = 0, False
StepIn_EStopMtr   = 0, False

CtrlIn_MotorMode  = 0, Disabled
CtrlIn_StepType   = 3, SixteenStep
CtrlIn_Direction  = 0, CounterCW
CtrlIn_AccTable   = 0, NoProfile
StepIn_PFDSetting = 0, Slow

CtrlIn_PhaseCurr  = 1230         [mA]
CtrlIn_NumOfSteps = 0
CtrlIn_StepRate   = 6.00         [St/s]

ErrorFlags        = 0X80000100
```

# 6. REFLASHING OVER CAN WITH THE AXIOMATIC EA BOOTLOADER

The AX100700 can be upgraded with new application firmware using the **Bootloader Information** section. This section details the simple step-by-step instructions to upload new firmware provided by Axiomatic onto the unit via CAN, without requiring it to be disconnected from the J1939 network.

*Note: To upgrade the firmware use Axiomatic Electronic Assistant V4.2.38.2 or higher.*

1. When the Axiomatic EA first connects to the ECU, the **Bootloader Information** section will display the following information.
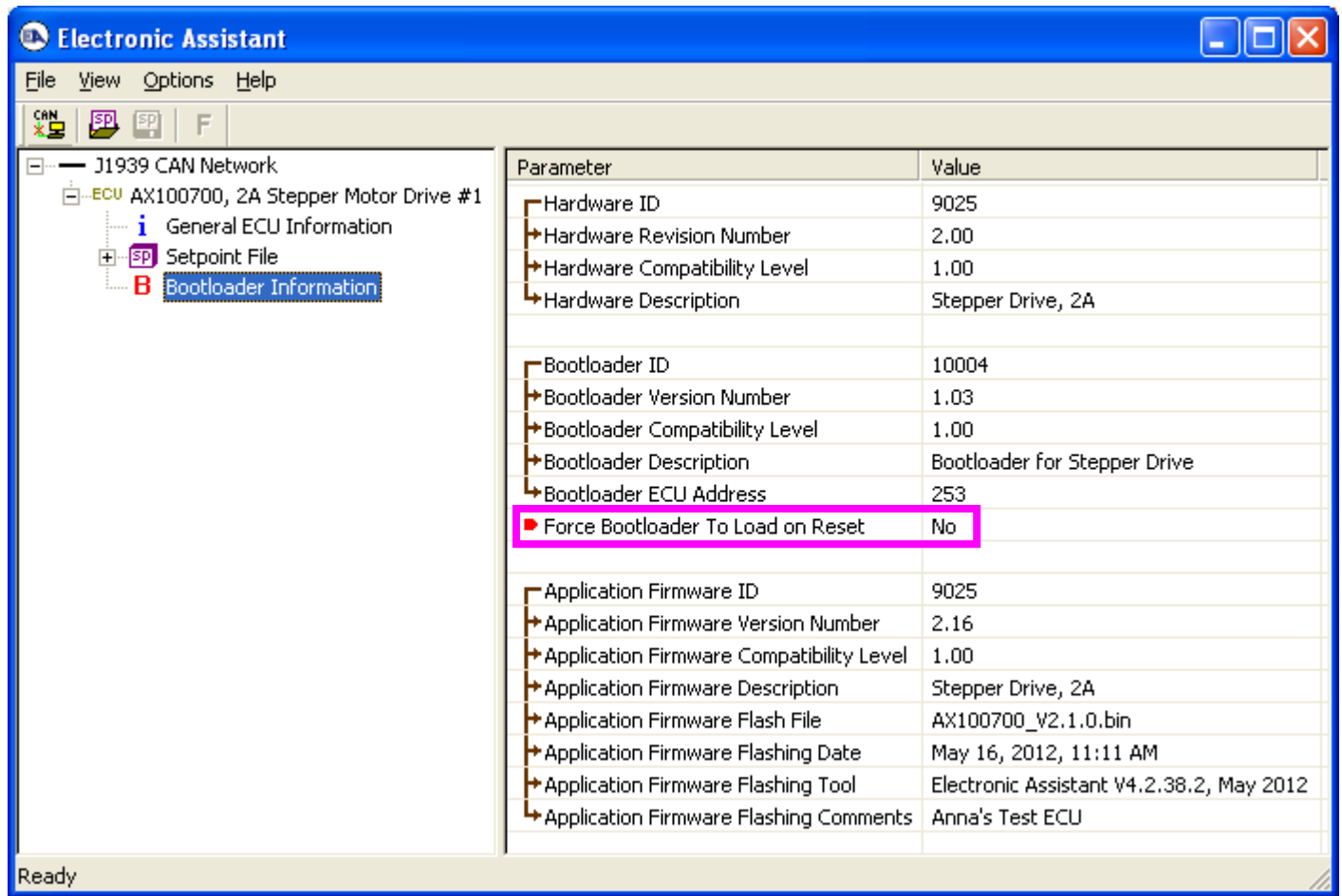
2. To use the bootloader to upgrade the firmware running on the ECU, change the variable **"Force Bootloader To Load on Reset"** to *Yes.*
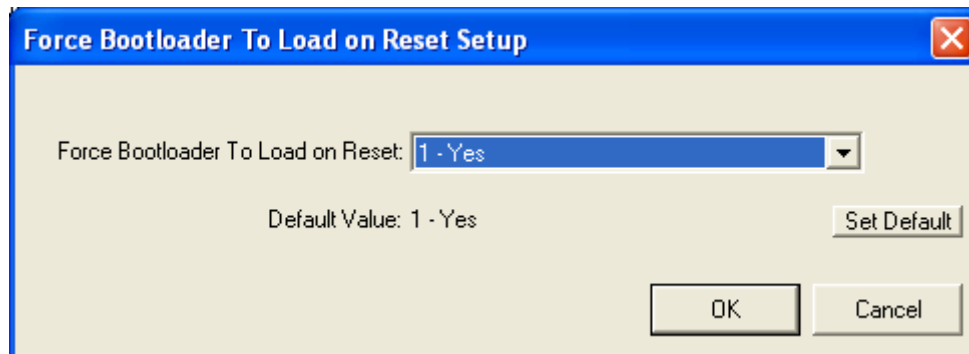
3. When the prompt box asks if you want to reset the ECU, select *Yes*.

Electronic Assistant

? Do you want to reset the ECU after changing this parameter ?

[Yes]   [No]
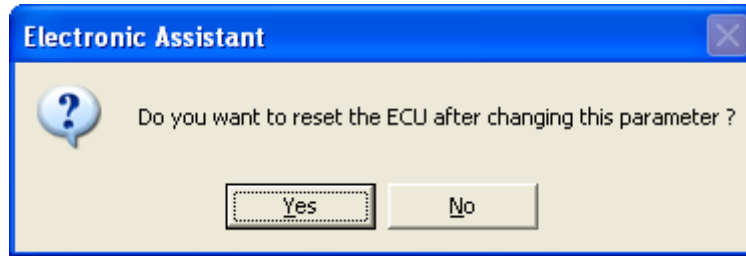
4. Upon reset, the ECU will no longer show up on the J1939 network as an AX022400 but rather as **J1939 Bootloader #1**.

Electronic Assistant

File   View   Options   Help

ECU	J1939 NAME	Address	J1939 Preferre...
ECU J1939 Bootloader #1	0X00FEFF00145D832B	0XFD	Reserved for OEM

⊟ — J1939 CAN Network
  ⊞ ECU J1939 Bootloader #1

Ready

Electronic Assistant

File   View   Options   Help

⊟ — J1939 CAN Network
  ⊟ ECU J1939 Bootloader #1
     i General ECU Information
     B Bootloader Information

Parameter	Value	Description
▶ ECU Part Number	AX100700	
▶ ECU Serial Number	0021710039	
⌐ECU J1939 NAME		PGN 60928. 64-bit ECU Identifier sent in Address
↦Arbitrary Address Capable	0X00	No
↦Industry Group	0X00	Global
↦Vehicle System Instance	0X00	
↦Vehicle System	0X7F	Not Available
↦Reserved	0X00	
↦Function	0XFF	Not Available
↦Function Instance	0X00	
↦ECU Instance	0X00	#1 - First Instance
↦Manufacturer Code	0X0A2	Axiomatic Technologies
↳Identity Number	0X1EB37E	Unique ECU network ID number
▶ ECU Address	0XFD	Reserved for OEM
—Software ID	N/A	PGN 65242 -SOFT

Ready

Note that the bootloader is NOT Arbitrary Address Capable. This means that if you want to have multiple bootloaders running simultaneously (not recommended) you would have to manually change the address for each one before activating the next, or there will be address conflicts, and only one ECU would show up as the bootloader. Once the 'active' bootloader returns to regular functionality, the other ECU(s) would have to be power cycled to re-activate the bootloader feature.

5. When the **Bootloader Information** section is selected, the same information is shown as when it was running the AX100700 firmware, but in this case the Flashing feature has been enabled.

6. Select the Flashing button and navigate to where you had saved the **AX100700_Vx.y.z.bin** file sent from Axiomatic. (Note: only binary (.bin) files can be flashed using the Axiomatic EA tool)

7. Once the Flash Application Firmware window opens, you can enter comments such as "Firmware upgraded by [Name]" if you so desire. This is not required, and you can leave the field blank if you do not want to use it.

Note: You do not have to date/timestamp the file, as this is done automatically by the Axiomatic EA tool when you upload the new firmware.
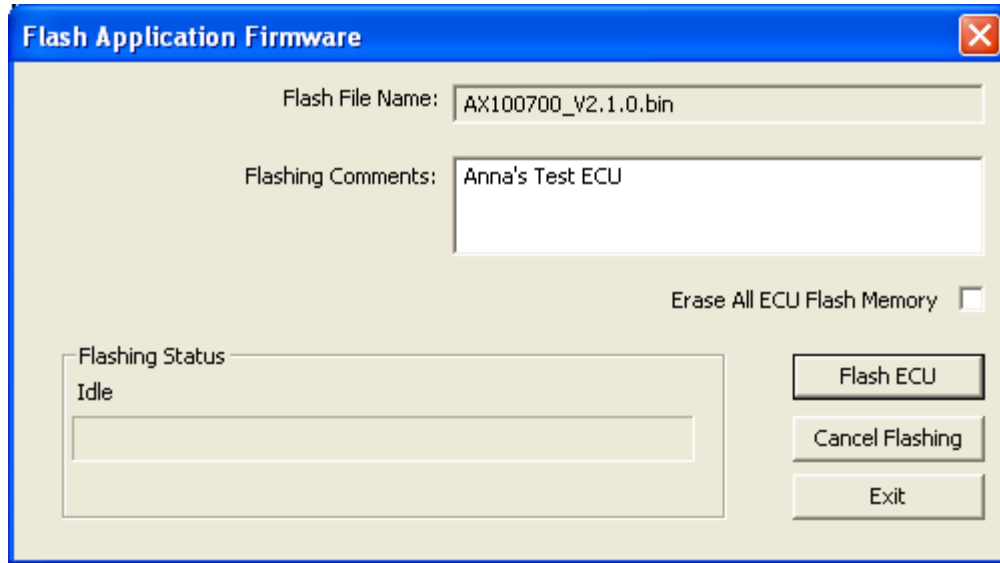

WARNING: Do not check the "Erase All ECU Flash Memory" box unless instructed to do so by your Axiomatic contact. Selecting this will erased ALL data stored in non-volatile flash, including the calibration done by Axiomatic during factory testing. It will also erase any configuration of the setpoints that might have been done to the ECU and reset all setpoints to their factory defaults. By leaving this box unchecked, none of the setpoints will be changed when the new firmware is uploaded.

8. A progress bar will show how much of the firmware has been sent as the upload progresses. The more traffic there is on the J1939 network, the longer the upload process will take.
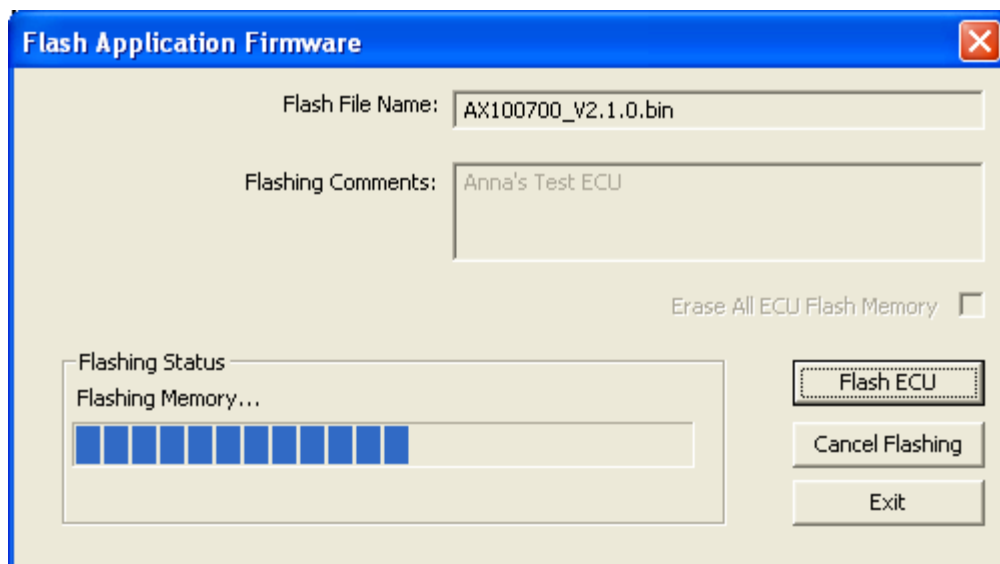
9. Once the firmware has finished uploading, a message will popup indicating the successful operation. If you select to reset the ECU, the new version of the AX100700 application will start running, and the ECU will be identified as such by the Axiomatic EA. Otherwise, the next time the ECU is power-cycled, the AX100700 application will run rather than the bootloader function.
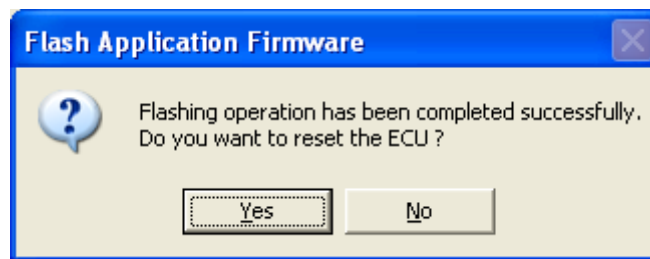
Note: If at any time during the upload the process is interrupted, the data is corrupted (bad checksum) or for any other reason the new firmware is not correct, i.e. bootloader detects that the file loaded was not designed to run on the hardware platform, the bad or corrupted application will not run. Rather, when the ECU is reset or power-cycled the **J1939 Bootloader** will continue to be the default application until valid firmware has been successfully uploaded into the unit.

APPENDIX A – Diagnostic Trouble Codes Table

J1939/73 DM1/2 SPN[1]/FMI	Lamp[1]	Description	Detection conditions	Associated Setpoint Group	Restricted Function
520193-05	Amber	Phase 1 Open Circuit	Average Motor Phase 1 Current < Phase Current Fault Threshold (when drive is ON and motor is stepping)	Phase 1 Winding Diag	Motor will not step
520194-05	Amber	Phase 2 Open Circuit	Average Motor Phase 2 Current < Phase Current Fault Threshold (when drive is ON and motor is stepping)	Phase 2 Winding Diag	Motor will not step
520195-31	Amber	Digital Output Open or Short Circuit	Hardware fault circuitry is triggered and detected by processor	Digital Output	None
520208-4	Amber	Voltage Input 1 Minimum Error	Measured Input < Minimum Error	Voltage/Current Input 1	Note 2
520208-3	Amber	Voltage Input 1 Maximum Error	Measured Input > Maximum Error	Voltage/Current Input 1	Note 2
520208-5	Amber	Current Input 1 Minimum Error	Measured Input < Minimum Error	Voltage/Current Input 1	Note 2
520208-6	Amber	Current Input 1 Maximum Error	Measured Input > Maximum Error	Voltage/Current Input 1	Note 2
520448-18	Amber	Frequency/PWM Input 1 Minimum Error	Measured Input < Minimum Error	Frequency/PWM Input 1	Note 2
520448-16	Amber	Frequency/PWM Input 1 Maximum Error	Measured Input > Maximum Error	Frequency/PWM Input 1	Note 2
524273-04	Amber	Power Supply Under Voltage	Measured Power Supply < Power Undervoltage Threshold	Power Supply Diag	Motor Off
524273-03	Amber	Power Supply Over Voltage	Measured Power Supply > Power Overvoltage Threshold	Power Supply Diag	Motor Off
524287-19	Amber	Lost Communication	Did not receive an expected CAN message within timeout period	CAN Receive X	Motor Off

Note 1: SPN, FMI and Lamp Type are user configurable parameters for all DTC. Default values are shown in this table

Note 2: An input used as a control or feedback input for a motor/output control block will disable the corresponding output when it is in an error state

Note 3: All diagnostic trouble code (DTC) can be disabled at any time by setting the "XXX Fault Generates DTCs" to FALSE.
    If this setpoint for all diagnostics are FALSE, the DM1 message will not be sent every 1 second (i.e. in a standalone drive mode where no CAN is connected)

APPENDIX B – Technical Specifications

Specifications are indicative and subject to change. Actual performance will vary depending on the application and operating conditions. Users should satisfy themselves that the product is suitable for use in the intended application. All our products carry a limited warranty against defects in material and workmanship. Please refer to our Warranty, Application Approvals/Limitations and Return Materials Process as described on https://www.axiomatic.com/service/
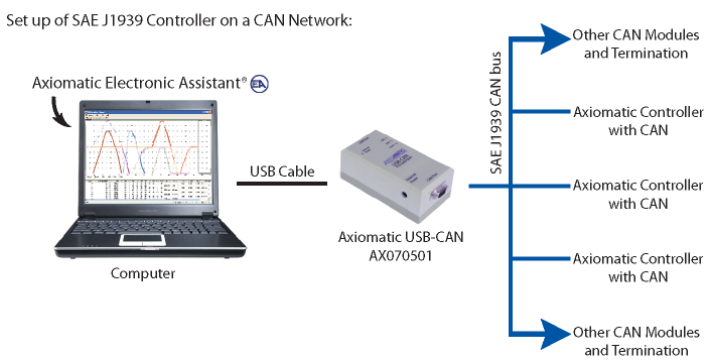
Input Specifications

Power Supply Input - Nominal	12V. 24V nominal 9…32 VDC
Surge Protection	Provided
Under-voltage Protection	Provided, software reaction
Over-voltage Protection	Provided, software reaction
Analog Input	1 analog input, for voltage or current measuring Can be configured as active high or low digital input Inputs are sampled every 1 msec. Protected against shorts to GND or +Vcc Accuracy: +/- 1% over range Resolution: 1mV or 1uA, depending on type
Frequency Input	1 Pulse or PWM, for inductive or active sensors User configurable for sampling range: 0.5 Hz to 50Hz; 10Hz to 1kHz; or 100Hz to 10 kHz Can be configured as active high or low digital input Inputs are sampled every 1 msec. Protected against shorts to GND or +Vcc Accuracy: +/- 1% over range for low frequency inputs (<1kHz) Resolution: 0.01 Hz or 0.1 Hz or 1Hz, depending on measuring range. 0.1% for PWM input
Signal Grounds	2 Analog GND (Input and +5Vref) 1 Frequency GND
Input Impedance	0-5V @ 1MOhm 0-20 mA @ 130 Ohms Frequency/PWM @ 10 KOhm pullup or pulldown

Output Specifications

Output to Motor	Allegro DMOS Microstepping Driver, A3979 Maximum 2A per phase for a Stepper Motor Some overcurrent protection is provided by the drive chip up to 2.5A. User has full access to all the chip's features via CAN messages.
Motor Disable	Motor can be disabled in a variety of ways, including "Motor Mode", and "E-Stop" input, or when an error is detected.
Motor Direction	Motor Direction can be user selected for Clockwise or Counter-Clockwise Drive. Refer to Section 1.2 for details.
Thermal Protection	Thermal protection is built-in to driver chip.
Digital Output	1 sinking (switched to GND) output for up to a 5A load
Reference Voltage	+5V, 10 mA is available to power a sensor or potentiometer and is referenced to Analog GND

General Specifications

Microprocessor	STM32F103, 32-bit at 72MHz
Efficiency	>95%
Motor Drive Logic	Standard embedded software is provided. The stepper drive is a highly programmable controller, allowing the user to configure it for their application. Its sophisticated control algorithms allow for open or closed loop drive of the motor. All I/O on the unit are inherently independent from one another, but can be programmed to interact in a large number of ways. All configurable parameters are user selectable using the Axiomatic Electronic Assistant. There are 2 types of function blocks associated with the stepper motor logic and drive. Stepper Drive Values: Default run-time values, ranges for linear controls Stepper Drive Controls: Selection of control sources for all the run-time variables associated with the stepper and output drive Up to 15 independent CAN Receive messages can be read from a J1939 network to control the various run-time variables associated with the stepper drive Up to 10 independent CAN Transmit messages can be sent to a J1939 network for system diagnostics and feedback
Diagnostics	Each input and output channel can be configured to send diagnostic messages to the CAN network if the I/O goes out of range. Diagnostic data is stored in a non-volatile log. Refer to Section 1.7 for details
Additional Fault Feedback	There are several types of faults that the controller will detect and provide a response: phase current winding problems; power supply undervoltage and overvoltage; and lost communication. They can be sent to the CAN bus.
CAN User Interface	Via The Axiomatic Electronic Assistant for *Windows* operating systems It comes with a royalty-free license for use. The Axiomatic Electronic Assistant requires an USB-CAN converter to link the device's CAN port to a *Windows*-based PC for initial configuration. An Axiomatic USB-CAN Converter AX070501 is available. Order the Axiomatic EA and USB-CAN as a kit (P/Ns: AX070502 or AX070506K), which includes all interconnecting cables.
RS-232	1 RS-232 port interfaces to a serial port (i.e. COM1) on a PC (115200 Baud Rate, N81, Xon/Xoff Flow Control). Not available through the connectors Connect the RS-232 port from the controller to a PC with Tera Term (Pro) freeware. The user can use the RS-232 port for diagnostics or testing. Changing setpoints is not possible through RS-232. The CAN port and the Axiomatic Electronic Assistant® must be used to configure the stepper drive. For further details refer to Section 5. NB. Tera Term (Pro) is freeware and is downloadable from http://hp.vector.co.jp/authors/VA002416/teraterm.html.
Electrical Connections	Refer to Section 2, Wires should be of the appropriate gauge to meet requirements of applicable electrical codes and suit the specifications of the connector(s).
Mounting	The controller has 2 mounting holes for a 0.25" (#6) fastener The bolt length will be determined by the end-user's mounting plate thickness. Typically 20 mm (3/4 inch) is adequate. The plastic housing provides limited EMI protection. For improved response, connect the shield pin on the connector to the chassis GND.
Packaging and Dimensions	NOTE: TE Deutsch connectors are rated at IP67 for submersion (3 ft., 0.9m) at IP69K for high pressure, high temperature wash down applications Refer to Figure 19 for dimensions
Operating Conditions	Operating: -40 to 85°C (-40 to 185°F)
Protection Rating	IP67 rating for product assembly
Regulatory Approvals	The stepper drive is not CE marked, as it is not a self-contained system.

OUR PRODUCTS

AC/DC Power Supplies

Actuator Controls/Interfaces

Automotive Ethernet Interfaces

Battery Chargers

CAN Controls, Routers, Repeaters

CAN/WiFi, CAN/Bluetooth, Routers

Current/Voltage/PWM Converters

DC/DC Power Converters

Engine Temperature Scanners

Ethernet/CAN Converters, Gateways, Switches

Fan Drive Controllers

Gateways, CAN/Modbus, RS-232

Gyroscopes, Inclinometers

Hydraulic Valve Controllers

Inclinometers, Triaxial

I/O Controls

LVDT Signal Converters

Machine Controls

Modbus, RS-422, RS-485 Controls

Motor Controls, Inverters

Power Supplies, DC/DC, AC/DC

PWM Signal Converters/Isolators

Resolver Signal Conditioners

Service Tools

Signal Conditioners, Converters

Strain Gauge CAN Controls

Surge Suppressors

OUR COMPANY

Axiomatic provides electronic machine control components to the off-highway, commercial vehicle, electric vehicle, power generator set, material handling, renewable energy and industrial OEM markets. *We innovate with engineered and off-the-shelf machine controls that add value for our customers.*

QUALITY DESIGN AND MANUFACTURING
We have an ISO9001:2015 registered design/manufacturing facility in Canada.

WARRANTY, APPLICATION APPROVALS/LIMITATIONS
Axiomatic Technologies Corporation reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. Users should satisfy themselves that the product is suitable for use in the intended application. All our products carry a limited warranty against defects in material and workmanship. Please refer to our Warranty, Application Approvals/Limitations and Return Materials Process at https://www.axiomatic.com/service/.

COMPLIANCE
Product compliance details can be found in the product literature and/or on axiomatic.com. Any inquiries should be sent to sales@axiomatic.com.

SAFE USE
All products should be serviced by Axiomatic. Do not open the product and perform the service yourself.

> ⚠️ This product can expose you to chemicals which are known in the State of California, USA to cause cancer and reproductive harm. For more information go to www.P65Warnings.ca.gov.

SERVICE
All products to be returned to Axiomatic require a Return Materials Authorization Number (RMA#) from sales@axiomatic.com. Please provide the following information when requesting an RMA number:
- Serial number, part number
- Runtime hours, description of problem
- Wiring set up diagram, application and other comments as needed

DISPOSAL
Axiomatic products are electronic waste. Please follow your local environmental waste and recycling laws, regulations and policies for safe disposal or recycling of electronic waste.

CONTACTS

Axiomatic Technologies Corporation
1445 Courtneypark Drive E.
Mississauga, ON
CANADA L5T 2E3
TEL: +1 905 602 9270
FAX: +1 905 602 9279
www.axiomatic.com
sales@axiomatic.com

Axiomatic Technologies Oy
Höytämöntie 6
33880 Lempäälä
FINLAND
TEL: +358 103 375 750
www.axiomatic.com
salesfinland@axiomatic.com