

# **UNIVERSAL INPUT, SINGLE OUTPUT VALVE CONTROLLER with LED**

**SAEJ1939®**

## **USER MANUAL**

**P/N: AX021610 (250kbps)**

**P/N: AX021610-01 (500kbps)**

**P/N: AX021610-02 (1Mbps)**

## ACCRONYMS

ACK	Positive Acknowledgement (from SAE J1939 standard)
UIN	Universal Input
LED	Light Emitting Diode
EA	The Axiomatic Electronic Assistant (A Service Tool for Axiomatic ECUs)
ECU	Electronic Control Unit (from SAE J1939 standard)
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
SPN	Suspect Parameter Number (from SAE J1939 standard)

**Note:**

An Axiomatic Electronic Assistant KIT may be ordered as P/N: AX070502 or AX070506K

## TABLE OF CONTENTS

<b>1.</b>	<b>OVERVIEW OF CONTROLLER</b>	<b>5</b>
1.1.	DESCRIPTION OF SINGLE UNIVERSAL INPUT TO PROPORTIONAL VALVE OUTPUT CONTROLLER	5
1.2.	DUAL LED OUTPUT FUNCTION BLOCK	6
1.3.	UNIVERSAL INPUT FUNCTION BLOCK	7
1.3.1.	Input Sensor Types	7
1.3.2.	Pullup / Pulldown Resistor Options	8
1.3.3.	Active High / Active Low Options	8
1.3.4.	Counter Type	9
1.3.5.	Minimum and Maximum Ranges	9
1.3.6.	Input Software Filter Types	10
1.4.	INTERNAL FUNCTION BLOCK CONTROL SOURCES	10
1.5.	DUAL LED BLOCK CONTROL SOURCES	12
1.6.	OUTPUT DRIVE FUNCTION BLOCKS	13
1.7.	PID CONTROL FUNCTION BLOCK	18
1.8.	DIAGNOSTIC FUNCTION BLOCKS	20
1.9.	SIMPLE CONDITIONAL LOGIC FUNCTION BLOCKS	24
1.10.	SET / RESET FUNCTION BLOCK	25
1.11.	LOOKUP TABLE FUNCTION BLOCK	26
1.11.1.	Auto Update on Setpoint Changes	26
1.11.2.	X-Axis, Input Data Response	26
1.11.3.	Y-Axis, Lookup Table Output	26
1.11.4.	Default Configuration, Data Response	27
1.11.5.	Point To Point Response	27
1.11.6.	X-Axis, Time Response	29
1.12.	PROGRAMMABLE LOGIC FUNCTION BLOCK	30
1.12.1.	Conditions Evaluation	33
1.12.2.	Table Selection	34
1.12.3.	Logic Block Output	35
1.13.	MATH FUNCTION BLOCK	36
1.14.	CAN TRANSMIT FUNCTION BLOCK	37
1.15.	DIAGNOSTIC TROUBLE CODE (DTC) REACT	38
1.16.	CAN RECEIVE FUNCTION BLOCK	38
<b>2.</b>	<b>INSTALLATION INSTRUCTIONS</b>	<b>39</b>
2.1.	DIMENSIONS AND PINOUT	39
2.2.	MOUNTING INSTRUCTIONS	39
<b>3.</b>	<b>OVERVIEW OF J1939 FEATURES</b>	<b>41</b>
3.1.	INTRODUCTION TO SUPPORTED MESSAGES	41
3.2.	NAME, ADDRESS AND SOFTWARE ID	42
<b>4.</b>	<b>ECU SETPOINTS ACCESSED WITH THE AXIOMATIC ELECTRONIC ASSISTANT</b>	<b>44</b>
4.1.	J1939 NETWORK	44
4.2.	MISCELLANEOUS SETPOINTS	44
4.3.	LED CONTROL	45
4.4.	UNIVERSAL INPUT	46
4.5.	PROPORTIONAL OUTPUT DRIVE	47
4.6.	CONSTANT DATA LIST SETPOINTS	49
4.7.	LOOKUP TABLE SETPOINTS	49
4.8.	PROGRAMMABLE LOGIC SETPOINTS	50
4.9.	MATH FUNCTION BLOCK SETPOINTS	52
4.10.	CAN RECEIVE SETPOINTS	53
4.11.	DTC REACT	54

- 4.12. **DIAGNOSTICS BLOCKS ..... 54**
- 4.13. **CAN TRANSMIT SETPOINTS..... 57**
- 5. REFLASHING OVER CAN WITH THE AXIOMATIC EA BOOTLOADER..... 59**
- 6. TECHNICAL SPECIFICATIONS..... 63**
  - 6.1. **POWER SUPPLY ..... 63**
  - 6.2. **INPUTS..... 63**
  - 6.3. **OUTPUT ..... 63**
  - 6.4. **COMMUNICATION..... 63**
  - 6.5. **GENERAL SPECIFICATIONS ..... 64**
- 7. VERSION HISTORY..... 65**

# 1. OVERVIEW OF CONTROLLER

## 1.1. Description of Single Universal Input to Proportional Valve Output Controller

This User Manual describes the architecture and functionality of the Universal Input to Single Output Valve Controller with LED.

All inputs and logical function blocks on the unit are inherently independent from one another, but can be configured to interact in a large number of ways. Figure 1 shows the hardware features of the 1IN-1OUT-LED.

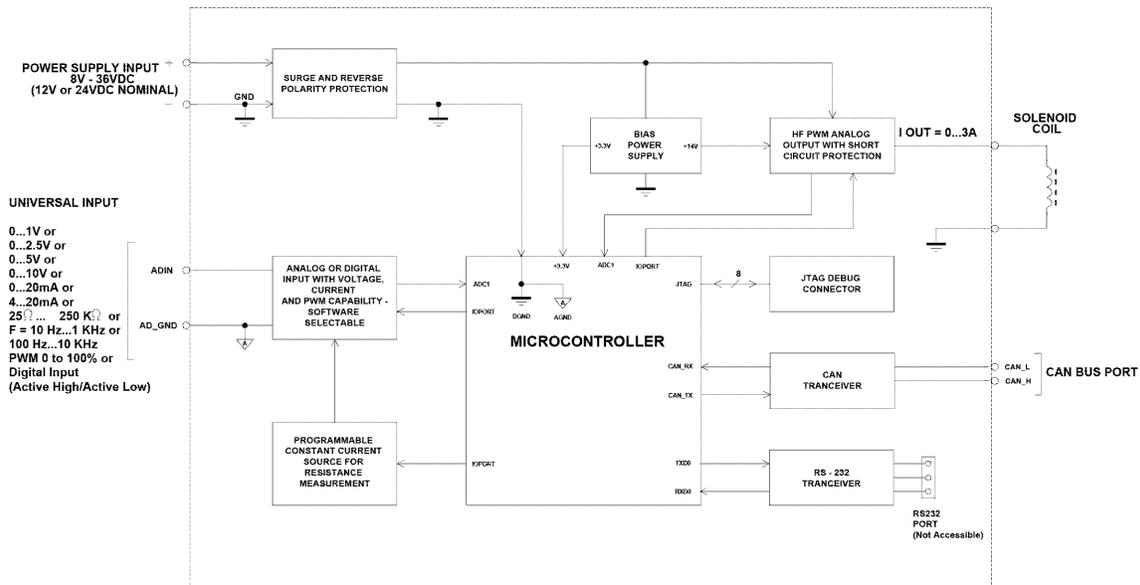


Figure 1 – Hardware Functional Block Diagram

The controller (1IN-1OUT-LED) is designed for versatile control of a universal input and a proportional valve output. Its flexible hardware design allows for the controller to have a wide range of input and output types. The sophisticated control algorithms/logical function blocks allow the user to configure the controller for a wide range of applications without the need for custom firmware.

The various function blocks supported by the 1IN-1OUT-LED are outlined in the following sections. All setpoints are configurable using Axiomatic service tool, the Axiomatic Electronic Assistant (EA).

The universal input can be configured to read analog signals: *Voltage, Current, and Resistance* as well as digital signals: *Frequency/RPM, PWM, Digital, and Counter types*. The inputs are described in more detail in section 1.3.

Similarly, the output can be configured to different types: *Proportional Current, Voltage, PWM, Hotshot Digital Current and Digital (ON/OFF)*. Each output consists of a high side half-bridge driver able to source up to 3Amps with hardware shutdown at 4Amps. The outputs are described in more detail in section 1.5.

Additionally, the 1IN-1OUT-LED includes a dual LED which is visible from outside the housing. The LED can be configured in various ways to visually inform the user of the controller's operations. The LED configuration is described in more detail in section 1.2.

## 1.2. Dual LED Output Function Block

The 1IN-1OUT-LED supports a dual LED (Green and Red). The dual LED output can be used to inform the user of the status the controller is in.

The structure of the output block for the dual LED output is based on stages. The 1IN-1OUT-LED provides the user with up to four different stages in order to provide more flexibility to the dual LED output functionality.

Each of the four stages consists of its independent control source and digital response. However, only one stage can be active at a time. For this reason, the 4-stage structure is based on priority in which the 4<sup>th</sup> stage has the highest priority and the 1<sup>st</sup> stage has the lowest priority. In other words: if all 4 stages are true, the response set for stage 4 will be used to drive the LED. In the instance when none of the stages is active (ON), the Dual LED function block has another group which is the default stage. When none of the stages are active, the user can configure the default stage to command the LED in various ways as long as none of the stages are active.

The control sources that can be used to command each of the four stages are listed in section 1.5 Table 11. The control sources available for the dual LED output consist of the same control source options as Table 9 with the addition of controller fault options to drive the LED. Hence, the dual LED can be configured to be driven when a (selected) fault occurs.

The output types available for the dual LED for each stage are listed in Table1 below

<b>Value</b>	<b>Meaning</b>
0	<i>LEDs Disabled</i>
1	<i>Green</i>
2	<i>Yellow</i>
3	<i>Red</i>
4	<i>Toggle Green/Red</i>

**Table 1 – LED Output Types**

The LED Responses that are available for each stage in the 1IN-1OUT-LED are listed in Table 2 below

<b>Value</b>	<b>Meaning</b>
0	<i>Normal On/Off</i>
1	<i>Blinking Logic</i>

**Table 2 – LED Response Options**

In 'Normal On/Off' response type, the output command state will follow the control input command: when the control input command is ON, the output will be turned ON and vice-versa.

In 'Blinking Logic' response type, the output will toggle at a period of '**Digital Blink Rate**' for as long as the input command is ON.

The "**Control Source**" setpoint together with "**Control Number**" setpoint determine which signal is used to drive the output. For example setting "**Control Source**" to '*Universal Input Measured*' and "**Control Number**" to '1', connects signal measured from Universal Input 1 to the output in question. The input signal is scaled per input type range between 0 and 1 to form control signal.

### 1.3. Universal Input Function Block

The controller consists of one universal input. The universal input can be configured to measure voltage, current, resistance, frequency, pulse width modulation (PWM) and digital signals.

#### 1.3.1. Input Sensor Types

Table 3 lists the supported input types by the controller. The **Input Sensor Type** parameter provides a dropdown list with the input types described in Table 3. Changing the **Input Sensor Type** affects other setpoints within the same setpoint group such as Minimum/Maximum Error/Range by refreshing them to new input type and thus should be changed first.

0	<i>Disabled</i>
10	<i>Voltage 0 to 1V</i>
11	<i>Voltage 0 to 2.5V</i>
12	<i>Voltage 0 to 5V</i>
13	<i>Voltage 0 to 10V</i>
20	<i>Current 0 to 20mA</i>
21	<i>Current 4 to 20mA</i>
30	<i>Resistive 30Ohm to 250kOhm</i>
40	<i>Frequency 0.5 to 50Hz</i>
41	<i>Frequency 10Hz to 1kHz</i>
42	<i>Frequency 100Hz to 10kHz</i>
50	<i>PWM Low Frequency (&lt;1kHz)</i>
51	<i>PWM High Frequency (&gt;100Hz)</i>
60	<i>Digital (Normal)</i>
61	<i>Digital (Inverse)</i>
62	<i>Digital (Latched)</i>
70	<i>Counter (0-10kHz)</i>

**Table 3 – Universal Input Sensor Type Options**

All analog input types are fed directly into a 12-bit analog-to-digital converter (ADC) in the microcontroller. All voltage input types are high impedance while current input types use a 124Ω resistor to measure the signal.

Frequency/RPM, Pulse Width Modulated (PWM) and Counter **Input Sensor Types** are connected to the microcontroller timers. **Pulses per Revolution** setpoint is only taken into consideration when

the **Input Sensor Type** selected is one of the frequency types as per Table 3. When **Pulses per Revolution** setpoint is set to 0, the measurements taken will be in units of [Hz]. If **Pulses per Revolution** setpoint is set to higher than 0, the measurements taken will be in units of [RPM].

Digital **Input Sensor Types** offers three modes: Normal, Inverse, and Latched. The measurements taken with digital input types are 1 (ON) or 0 (OFF).

### 1.3.2. Pullup / Pulldown Resistor Options

With **Input Sensor Types**: Frequency/RPM, PWM, Digital, the user has the option of three (3) different pull up/pull down options as listed in Table 4.

0	<i>Pullup/Pulldown Off</i>
1	<i>10kΩ Pullup</i>
2	<i>10kΩ Pulldown</i>

Table 4 – Pullup/Pulldown Resistor Options

These options can be enabled or disabled by adjust the setpoint **Pullup/Pulldown Resistor** in the Axiomatic Electronic Assistant.

### 1.3.3. Active High / Active Low Options

In the case of Digital Inputs, the options **Active High/Active Low** are used to configure how a signal high or signal low is interpreted. The following table below are the available options provided. By default, *Active High* is selected.

0	<i>Active High</i>
1	<i>Active Low</i>

Table 5 – Active High / Active Low Options

Table 6 show the effect of different digital input types on input signal measurement interpretation with recommended **Pullup/Pulldown Resistor** and **Active High/Active Low** combinations. Fault diagnostics are not available in Digital Input types.

<b>Input Sensor Type</b>		<b><i>Active High</i> with 10kΩ Pulldown</b>	<b><i>Active Low</i> with 10kΩ Pullup</b>	<b>Input State</b>
60	<i>Digital (Normal)</i>	Signal High	Signal Low or Open	1 (ON)
		Signal Low or Open	Signal High	0 (OFF)
61	<i>Digital (Inverse)</i>	Signal High or Open	Signal Low	1 (ON)
		Signal Low	Signal High or Open	0 (OFF)
62	<i>Digital (Latched)</i>	Signal High to Low	Signal Low to High	0 (No Change)
		Signal Low to High	Signal High to Low	1 (State Change)

Table 6 – Digital Input State options based on Selected Options

### 1.3.4. Counter Type

The 1IN-1OUT-LED controller supports three different variations of the Counter input type. These variations are listed in Table 7 below:

0	<i>Pulses Within Measuring Window</i>
1	<i>Time Measurement of Pulse Count</i>
2	<i>Trigger on Pulse Count Completion</i>

Table 7 – Counter Input Types

The first option *Pulses Within Measuring Window* is used to count the number of pulses that occur within a configurable time frame (Measuring Window) in terms of milliseconds. This particular Counter type uses the other setpoint **Measuring Window** to select the time frame in which pulses are to be counted.

*Time Measurement of Pulse Count* is another option which allows time measurement of selected number of pulses to occur. Time measurement can be from 0ms to 65,000ms. If the time measurement has reached its maximum but not yet counted the selected number of pulses, the time will remain at maximum value. Therefore, it is important to select a number of pulses which could occur within the maximum time frame. Once the selected number of pulses have been read, the time will be restarted until the pulses have been read.

*Trigger on Pulse Count Completion* is Counter input type which triggers an 'ON' signal as soon as the selected number of pulses have been read. Setpoints **Pulses to Count**, **Edge to Trigger Pulse** and **Trigger on First Pulse** work together in this particular Counter type. When the controller reads all pulses in **Pulses to Count**, the input will be triggered 'ON' until the next pulse is received which will reset the trigger to 'OFF'. The setpoint **Edge to Trigger** determines on which edge of the pulse is the pulse count to be incremented. If the edge selected is *Falling Edge*, then the pulse count will not be incremented until the falling edge of the pulse is received. Likewise, if the selected edge is *Rising Edge*, then the pulse count will be incremented as soon as the rising edge of the pulse is received. The other setpoint is **Trigger on First Pulse**. This setpoint gives the user the option of triggering on the first pulse counted or the last pulse counted in **Pulses to Count**.

For example, if **Pulses to Count** is set to 100, **Edge to Trigger** is set to *Rising Edge* and **Trigger on First Pulse** is set to TRUE, then the rising edge of the first pulse will trigger an 'ON' signal. The signal will remain 'ON' until the rising edge of the second pulse is received which turns the signal to 'OFF'. After a count of another 100 pulses is met, the signal will turn back 'ON' and the cycle starts again.

### 1.3.5. Minimum and Maximum Ranges

The **Minimum Range** and **Maximum Range** setpoints are used to create the overall useful range of the inputs. For example if **Minimum Range** is set to 0.5V and **Maximum Range** is set to 4.5V, the overall useful range (0-100%) is between 0.5V to 4.5V. Anything below the **Minimum Range** will saturate at **Minimum Range**. Similarly, anything above the **Maximum Range** will saturate at **Maximum Range**. In order to generate an input fault if the measured input falls output of the **Minimum Range** or **Maximum Range**, the Diagnostics function block can be used. Please refer to Section 1.8 for more details.

### 1.3.6. Input Software Filter Types

All input types with the exception of Digital (Normal), Digital (Inverse), Digital (Latched) can be filtered using **Filter Type** and **Filter Constant** setpoints. There are three (3) filter types available as listed in Table 8.

0	<i>No Filtering</i>
1	<i>Moving Average</i>
2	<i>Repeating Average</i>

**Table 8 – Input Filtering Types**

The first filter option *No Filtering*, provides no filtering to the measured data. Thus the measured data will be directly used to the any function block which uses this data.

The second option, *Moving Average*, applies the ‘Equation 1’ below to measured input data, where  $Value_N$  represents the current input measured data, while  $Value_{N-1}$  represents the previous filtered data. The Filter Constant is the **Filter Constant** setpoint.

Equation 1 - Moving Average Filter Function:

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

The third option, *Repeating Average*, applies the ‘Equation 2’ below to measured input data, where N is the value of **Filter Constant** setpoint. The filtered input, Value, is the average of all input measurements taken in N (**Filter Constant**) number of reads. When the average is taken, the filtered input will remain until the next average is ready.

Equation 2 - Repeating Average Transfer Function:

$$Value = \frac{\sum_0^N Input_N}{N}$$

### 1.4. Internal Function Block Control Sources

The 1IN-1OUT-LED controller allows for internal function block sources to be selected from the list of the logical function blocks supported by the controller. As a result, any output from one function block can be selected as the control source for another. Keep in mind that not all options make sense in all cases, but the complete list of control sources is shown in Table 9.

<b>Value</b>	<b>Meaning</b>
0	<i>Control Source Not Used</i>
1	<i>CAN Receive Message</i>
2	<i>Universal Input Measured</i>
3	<i>Output Target</i>

4	<i>Output Feedback</i>
5	<i>Lookup Table Function Block</i>
6	<i>Mathematical Function Block</i>
7	<i>Programmable Logic Function Block</i>
8	<i>PID Function Block</i>
9	<i>Conditional Block</i>
10	<i>Set-Reset Block</i>
11	<i>Constant Data List Block</i>
12	<i>Diagnostic Trouble Code</i>
13	<i>Measured Power Supply</i>
14	<i>Measured Processor Temperature</i>
15	<i>Receive Message Timeout</i>

**Table 9 – Control Source Options**

In addition to a source, each control also has a number which corresponds to the sub-index of the function block in question. Table 10 outlines the ranges supported for the number objects, depending on the source that had been selected.

<b>Control Source</b>	<b>Control Source Number</b>
Control Source Not Used (Ignored)	[0]
CAN Receive Message	[1...5]
<i>Universal Input Measured</i>	[1...1]
<i>Output Target</i>	[1...1]
<i>Output Feedback</i>	[1...1]
Lookup Table Function Block	[1...5]
Mathematical Function Block	[1...4]
Programmable Logic Function Block	[1...2]
<i>PID Function Block</i>	[1...1]
Conditional Logic Block	[1...4]
<i>Set-Reset Block</i>	[1...3]
<i>Constant Data List Block</i>	[1...15]
Diagnostic Trouble Code	[1...5]
Measured Power Supply	[1...1]
Measured Processor Temperature	[1...1]
Receive Message Timeout	[1...1]

**Table 10 – Control Source Number Options**

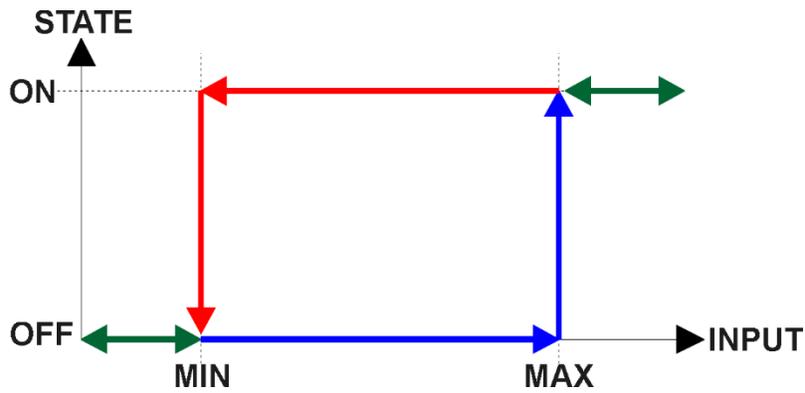


Figure 2 - Analog source to Digital input

### 1.5. Dual LED Block Control Sources

The Dual LED function block allows for the LEDs to be driven by the list shown in Section 1.4 as well as faults that have occurred in the controller. Table 11 lists all available control sources for the dual LED function block

Value	Meaning
0	Control Source Not Used
1	Universal Input Fault
2	Proportional Output Fault
3	CAN Communication Fault
4	Power Supply Fault
5	Temperature Fault
6	CAN Receive Message
7	Universal Input Measured
8	Output Target
9	Output Feedback
10	Lookup Table Function Block
11	Mathematical Function Block
12	Programmable Logic Function Block
13	PID Function Block
14	Conditional Block
15	Set-Reset Block
16	Constant Data List Block
17	Diagnostic Trouble Code

Table 11 – Control Source Options

In addition to a source, each control also has a number which corresponds to the sub-index of the function block in question. Table 12 outlines the ranges supported for the number objects, depending on the source that had been selected.

Control Source	Control Source Number
Control Source Not Used (Ignored)	[0]
<i>Universal Input Fault</i>	[1...1]
<i>Proportional Output Fault</i>	[1...1]
<i>CAN Communication Fault</i>	[1...1]
<i>Power Supply Fault</i>	[1...1]
<i>Temperature Fault</i>	[1...1]
CAN Receive Message	[1...5]
<i>Universal Input Measured</i>	[1...1]
<i>Output Target</i>	[1...1]
<i>Output Feedback</i>	[1...1]
Lookup Table Function Block	[1...5]
Mathematical Function Block	[1...4]
Programmable Logic Function Block	[1...2]
<i>PID Function Block</i>	[1...1]
Conditional Logic Block	[1...4]
<i>Set-Reset Block</i>	[1...3]
<i>Constant Data List Block</i>	[1...15]
Diagnostic Trouble Code	[1...5]

Table 12 – Control Source Number Options

## 1.6. Output Drive Function Blocks

The controller consists of a single proportional output. Output consists of a high side half-bridge driver able to source up to 3Amps. The outputs are connected to independent microcontroller timer peripherals and thus can be configured independently from 1Hz to 25kHz.

The Output Type setpoint determines what kind of signal the output produces. Changing this setpoint causes other setpoints in the group to update to match selected type. For this reason, the first setpoint that should be changed prior to configuring other setpoints is the Output Type setpoint. The supported output types by the controller are listed in Table 13 below:

0	Disabled
1	Proportional Current
2	Digital Hotshot
3	PWM Duty Cycle
4	Proportional Voltage (0-Vps)
5	Digital (0-Vps)

Table 13 – Output Type Options

There are two setpoints that are associated to *Proportional Current* and *Digital Hotshot Output Types* that are not with others - these are *Dither Frequency* and *Dither Amplitude*. The dither signal is used in *Proportional Current* mode and is a low frequency signal superimposed on top of the high frequency (25kHz) signal controlling the output current. The two outputs have independent dither frequencies which can be adjusted at any time. The combination of *Dither Amplitude* and *Dither Frequency* must be appropriately selected to ensure fast response to the coil to small changes in the control inputs but not so large as to affect the accuracy or stability of the output.

In *Proportional Voltage* type, the controller measures the Vps applied to the unit and based on this information, the controller will adjust the PWM duty cycle of the signal (0-Vps amplitude) so that the average signal is the commanded target value. Thus, the output signal is not an analog one. In order to create an analog signal, a simple low pass filter can be connected externally to the controller. Note: the output signal will saturate at Vps if the **Output At Maximum Command** is set higher than the supply voltage powering the controller.

In *PWM Duty Cycle Output Type*, the controller outputs a signal (0-Vps amplitude) on a fixed output frequency set by **PWM Output Frequency** with varying PWM Duty Cycle based on commanded input. Since both outputs are connected to independent timers, the **PWM Output Frequency** setpoint can be changed at any time for each output without affecting the other.

*Digital Output Type* offers the user with 4 different output responses as listed in Table 14. The controller will source any current required in any of the options listed in Table 14 up to 3Amps.

0	Normal On/Off
1	Inverse Logic
2	Latched Logic
3	Blinking Logic
4	Period Logic

**Table 14 – Output Type Options**

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.

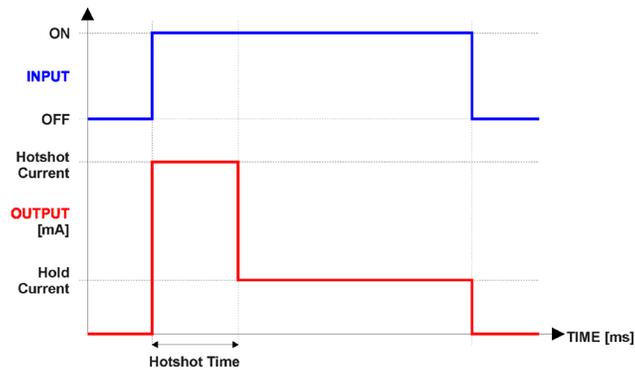
If a *Latched* response is selected, when the input commands the state from OFF to ON, the output will change state.

If a *Blinking* response is selected, then while the input command the output ON, it will blink at the rate in the **Digital Blink Rate** setpoint. When commanded OFF, the output will stay off. A blinking response is only available with a *Digital On/Off* type of output (not a Hotshot type.)

If a *Period* response is selected, then three setpoints work together with this option. If setpoint **Complete Full ON/OFF Cycle** is set to *FALSE*, then while the input command the output ON, the output will follow the **Digital ON Time** and **Digital OFF Time** periodically until the input command is OFF. On the other hand, if **Complete Full ON/OFF Cycle** is set to *TRUE*, then the output will complete only one (1) full cycle of the **Digital ON Time** and **Digital OFF Time**. In this case, if the input command signal remains ON after **Digital OFF Time** the output will remain off, until the command input has transitioned states from ON to OFF back to ON to complete another one (1) full cycle. Similarly, if the input command signal goes to OFF while the cycle is not completed, the output will continue the cycle until it has finished and wait until the input command transition to occur. The **Digital ON Time** and **Digital OFF Time** setpoints can be static or dynamic depending on the control source used.

The *‘Hotshot Digital’* type is different from *‘Digital On/Off’* in that it still controls the current through the load. This type of output is used to turn on a coil then reduce the current so that the valve will remain open, as shown in **Figure 3**. Since less energy is used to keep the output engaged, this type of response is very useful to improve overall system efficiency. With this output type there are

associated three setpoints: **Hold Current**, **Hotshot Current** and **Hotshot Time** which are used to configure form of the output signal as shown in Figure 3.



**Figure 3 – Hotshot Digital Profile**

For Proportional outputs signal minimum and maximum values are configured with **Output At Minimum Command** and **Output At Maximum Command** setpoints. Value range for both of the setpoints is limited by selected **Output Type**.

Regardless of what type of control input is selected, the output will always respond in a linear fashion to changes in the input per 'Equation 3'.

$$y = mx + a$$

$$m = \frac{Y_{max} - Y_{min}}{X_{max} - X_{min}}$$

$$a = Y_{min} - m * X_{min}$$

**Equation 3 - Linear Slope Calculations**

In the case of the Output Control Logic function block, X and Y are defined as

$X_{min}$  = Control Input Minimum ;  $Y_{min}$  = **Output at Minimum Command**

$X_{max}$  = Control Input Maximum;  $Y_{max}$  = **Output at Maximum Command**

In all cases, while X-axis has the constraint that  $X_{min} < X_{max}$ , there is no such limitation on the Y-axis. Thus configuring **Output At Minimum Command** to be greater than **Output At Maximum Command** allows output to follow control signal inversely.

In order to prevent abrupt changes at the output due to sudden changes in the command input, the user can choose to use the independent up or down ramps to smooth out the coil's response. The **Ramp Up** and **Ramp Down** setpoints are in milliseconds, and the step size of the output change will be determined by taking the absolute value of the output range and dividing it by the ramp time.

The output of the controller is factory calibrated according to the output regulator parameters set by default by Axiomatic. These parameters are: **Output Regulator Proportional Gain**, **Output Regulator Integral Gain**, **Output Regulator Derivative Gain**. However, depending on an

application, it may be necessary to tune these values to achieve a certain waveform or amplitude, especially if dither is used and superimposed onto the output current signal.

The **Control Source** setpoint together with **Control Number** setpoint determine which signal is used to drive the output. For example setting **Control Source** to *Universal Input Measured* and **Control Number** to (1) will connect signal measured from Universal Input1 to the output in question. The input signal is scaled per input type range between 0 and 1 to form control signal. Outputs respond in a linear fashion to changes in control signal. If a non-digital signal is selected to drive digital output the command state will be 0 (OFF) at or below the “**Output At Minimum Command**”, 1 (ON) at or above “**Output At Maximum Command**” and will not change in between those points.

In addition to the **Control Source** setpoint, the controller offers two more options that help increase its versatility – **Enable Source/Number/Response** and **Override Source/Number/Response** set of setpoints.

The **Enable Source** setpoint together with **Enable Number** setpoint determine the enable signal for the output in question. The **Enable Response** setpoint is used to select how output will respond to the selected Enable signal. **Enable Response** setpoint options are listed in Table 15.

0	Enable When On, Else Shutoff
1	Enable When On, Else Rampoff
2	Enable When On, Else Ramp To Max
3	Enable When On, Else Ramp To Min
4	Enable When On, Else Keep Last Value
5	Enable When Off, Else Shutoff
6	Enable When Off, Else Rampoff
7	Enable When Off, Else Ramp To Max
8	Enable When Off, Else Ramp To Min
9	Enable When Off, Else Keep Last Value

**Table 15 – Enable Response Options**

Override input allows the output drive to be configured to go to a default value in the case of the override input being engaged or disengaged, depending on the logic selected in **Override Response**, presented on Table 16. When active, the output will be driven to the value in **Output at Override Command** regardless of the value of the Control input. The **Override Source** and **Override Number** together determine the Override input signal.

0	Override When On
1	Override When Off

**Table 16 – Override Response Options**

If a fault is detected in any of the active inputs (Control/Enable/Override) the output will respond per **Control Fault Response** setpoint as outlined in Table 17. Fault Value is defined by **Output in Fault Mode** setpoint value, which is interpreted in selected output units.

0	Shutoff Output
1	Apply Fault Value
2	Hold Last Value

**Table 17 – Fault Response Options**

Besides Enable and Override signals controlling a particular output; other fault modes than can occur are those of a Power Supply over/under voltage and Microcontroller over-temperature. When any of these faults occur, the output is automatically disabled until the power supply or temperature are back in proper operating range. Refer to Section 1.8 for more details.

The outputs are inherently protected against a short to GND or Vps by hardware. In case of a dead short, the hardware will automatically disable the output drive, regardless of what the processor is commanding for the output. When this happens, the processor detects output hardware shutdown and commands off the output in question. It will continue to drive non-shortened outputs normally and periodically try to re-engage the short load, if still commanded to do so. If the fault has gone away since the last time the output was engaged while shorted, the controller will automatically resume normal operation.

In the case of an open circuit, there will be no interruption of the control for any of the outputs. The processor will continue to attempt to drive the open load.

The measured current through the load is available to be broadcasted on a CAN message if desired. It is also used as the input to the diagnostic function block for each output, and an open or shorted output can be broadcasted in a DM1 message on the CAN network

## 1.7. PID Control Function Block

The PID Control function block is an independent logic block, but it is normally intended to be associated with proportional output control blocks described earlier. When the **Control Source** for an output has been setup as a 'PID Function Block', the command from the selected PID block drives the physical output on the 1IN-1OUT-LED controller. The **PID Target Command Source** and **PID Target Command Number** setpoints determine control input and the **PID Feedback Input Source** and **PID Feedback Input Number** setpoints determine the established the feedback signal to the PID function block. The **PID Response Profile** will use the selected inputs as per the options listed in Table 18. When active, the PID algorithm will be called every **PID Loop Update Rate** in milliseconds.

0	Single Output
1	Setpoint Control
2	On When Over Target
3	On When Below Target

Table 18 – PID Response Options

When a '*Single Output*' response is selected, the Target and Feedback inputs do not have to share the same units. In both cases, the signals are converted to a percentage values based on the minimum and maximum values associated with the source function block.

For example, a CAN command could be used to set the target value, in which case it would be converted to a percentage value using **Receive Data Min** and **Receive Data Max** setpoints in the appropriate *CAN Receive X* function block. The closed-loop feedback signal (i.e. a 0-5V input) could be connected to '*Universal Input 1*' and selected as the feedback source. In this case the value of the input would be converted to a percentage based on the **Minimum Range** and **Maximum Range** setpoints in the input block. The output of the PID function would depend on the difference between the commanded target and the measured feedback as a percentage of each signals range. In this mode, the output of the block would be a value from -100% to 100%.

When a *Setpoint Control* response is selected, the **PID Target Command Source** automatically gets updated to *Control Constant Data* and cannot be changed. The value set in the associated constant in the Constant Data List function block becomes the desired target value. In this case, both the target and the feedback values are assumed to be in same units and range. The minimum and maximum values for the feedback automatically become the constraints on the constant target. In this mode, the output of the block would be a value from 0% to 100%.

For example, if the feedback was setup as a 4-20mA input, a **Constant Value X** setpoint set to 14.2 would automatically be converted to 63.75%. The PID function would adjust the output as needs to have the measured feedback to maintain that target value.

The last two response options, '*On When Over Target*' and '*On When Under Target*', are designed to allow the user to combine the two proportional outputs as a push-pull drive for a system. Both outputs must be setup to use the same control input (linear response) and feedback signal in order to get the expected output response. In this mode, the output would be between 0% and 100%.

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 below will be set to zero.

The PID algorithm used is shown below, where  $G$ ,  $K_i$ ,  $T_i$ ,  $K_d$ ,  $T_d$  and  $Loop\_Update\_Rate$  are configurable parameters.

$$PIDOutput_k = P_k + I_k + D_k$$

$$P_k = P\_Gain * Error_k$$

$$I_k = I\_Gain * ErrorSum_k$$

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

$$Error_k = Target - Feedback$$

$$ErrorSum_k = ErrorSum_{k-1} + Error_k$$

$$P\_Gain = G$$

$$I\_Gain = G * K_i * T / T_i \text{ (Note: If } T_i \text{ is zero, } I\_Gain = 0)$$

$$D\_Gain = G * K_d * T_d / T$$

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

#### Equation 4 - PID Control Algorithm

Each system will have to be turned for the optimum output response. Response times, overshoots and other variables will have to be decided by the customer using an appropriate PID tuning strategy. Axiomatic is not responsible for tuning the control system.

## 1.8. Diagnostic Function Blocks

The 1IN-1OUT-LED supports diagnostic messaging. DM1 message is a message, containing Active Diagnostic Trouble Codes (DTC) that is sent to the J1939 network in case a fault has been detected. A Diagnostic Trouble Code is defined by the J1939 standard as a four byte value.

In addition to supporting the DM1 message, the following are supported:

DM2	Previously Active Diagnostic Trouble Codes	Sent only on request
-----	--	----------------------

SPN	Suspect Parameter Number	(user defined)
FMI	Failure Mode Identifier	(see 0 and Table 11)
CM	Conversion Method	(always set to 0)
OC	Occurrence Count	(number of times the fault has happened)
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

Fault detection and reaction is a standalone functionality that can be configured to monitor and report diagnostics of various controller parameters. The 1IN-1OUT-LED supports 8 Diagnostics Definitions, each freely configurable by the user.

By default, the monitoring of operating voltage, CPU temperature and receive message timeouts is configured to diagnostics blocks 1, 2 and 3., In case any of these three diagnostics blocks are needed for some other use, the default settings can be adjusted by the user to suit the application.

There are 4 fault types that can be used, “**Minimum and maximum error**”, “**Absolute value error**”, “**State error**” and “**Double minimum and maximum error**”.

**Minimum and maximum error** has two thresholds, “MIN Shutdown” and “MAX Shutdown” that have configurable, independent diagnostics parameters (SPN, FMI, Generate DTCs, delay before flagging status). In case the parameter to monitor stays between these two thresholds, the diagnostic is not flagged.

**Absolute value error** has one configurable threshold with configurable parameters. In case the parameter to monitor stays below this threshold, the diagnostic is not flagged.

**State error** is similar to the Absolute value error, the only difference is that State error does not allow the user to specify specific threshold values; thresholds ‘1’ and ‘0’ are used instead. This is ideal for monitoring state information, such as received message timeouts.

**Double minimum and maximum error** lets user to specify four thresholds, each with independent diagnostic parameters. The diagnostic status and threshold values is determined and expected as show in Figure 4 below.

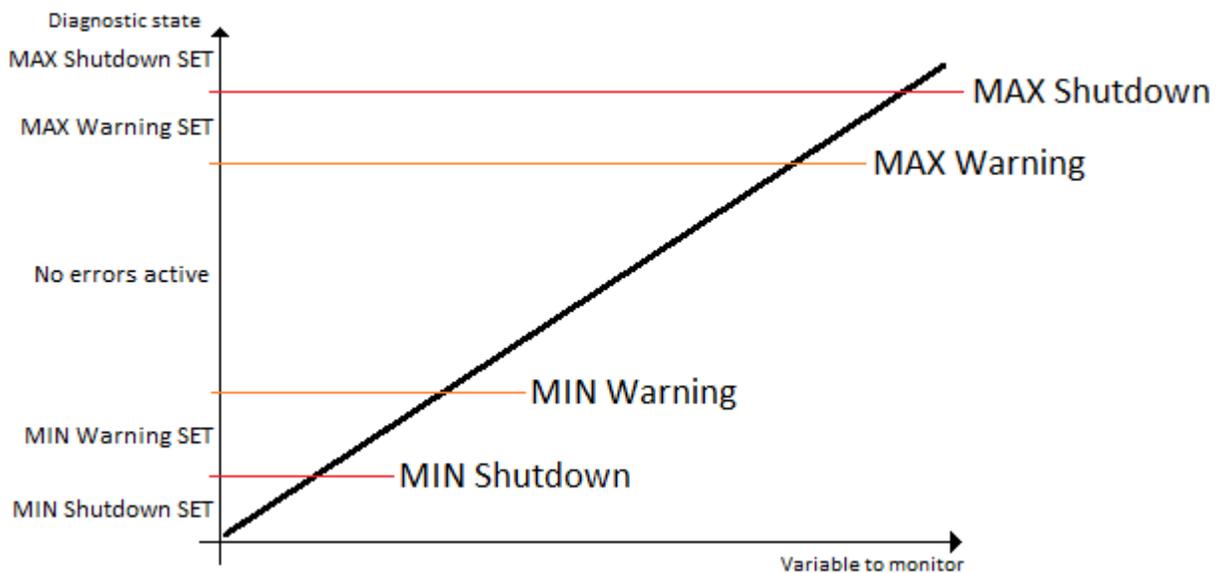


Figure 4 – Double Minimum and Maximum Error Thresholds

In case any of the Diagnostics blocks is configured to monitor Output Current Feedback, there is an internal error status flag maintained automatically for that particular output. This internal flag can be used for driving the particular output to a specified state in case of diagnostic event using Proportional Current Output setpoints “Control Fault Response”, “Output in Fault Mode” and “Fault Detection Enabled”.

There is also built in error status flags for power supply and CPU temperature monitoring. In case any of the diagnostics blocks is measuring these two parameters, the corresponding internal error status flags can be used for shutting down the unit in case of failure. The setpoints **Power Fault Disables Output** and **Over Temperature Shutdown** can be used for enabling the shutdown of the unit (shutdown == output driving is turned off).

While there are no active DTCs, the 1IN-1OUT-LED will send “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, a DM1 indicating that there are no more active DTCs will be sent.

If there is more than one active DTC at any given time, the regular DM1 message will be sent using a multipacket message to the Requester Address using the Transport Protocol (TP).

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 Event is flagged**” 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.

By default, the fault flag is cleared when error condition that has caused it goes away. The DTC is made Previously Active and is no longer included in the DM1 message. To identify a fault having happened, even if the condition that has caused is one away, the “**Event Cleared only by DM11**” setpoint can be set to ‘True’. This configuration enables DTC to stay Active, even after the

fault flag has been cleared, and be included in DM1 message until a Diagnostic Data Clear/Reset for Active DTCs (DM11) has been requested.

As defined by J1939 Standard the first byte of the DM1 message reflects the Lamp status. **“Lamp Set by Event”** setpoint determines the lamp type set in this byte of DTC. **“Lamp Set by Event”** setpoint options are listed in Table 19. By default, the ‘*Amber, Warning*’ lamp is typically the one set be any active fault.

Value	Meaning
0	Protect
1	Amber Warning
2	Red Stop
3	Malfunction

**Table 19 – Lamp Set by Event in DM1 Options**

**“SPN for Event”** defines suspect parameter number used as part of DTC. The default value zero is not allowed by the standard, thus no DM will be sent unless **“SPN for Event”** in is configured to be different from zero. **It is user’s responsibility to select SPN that will not violate J1939 standard.** When the **“SPN for Event”** is changed, the OC of the associated error log is automatically reset to zero.

Value	Meaning
0	<i>Data Valid But Above Normal Operational Range - Most Severe Level</i>
1	<i>Data Valid But Below Normal Operational Range - Most Severe Level</i>
2	<i>Data Intermittent</i>
3	<i>Voltage Above Normal, Or Shorted To High Source</i>
4	<i>Voltage Below Normal, Or Shorted To Low Source</i>
5	<i>Current Below Normal Or Open Circuit</i>
6	<i>Current Above Normal Or Grounded Circuit</i>
7	<i>Mechanical Error</i>
8	<i>Abnormal Frequency Or Pulse Width Or Period</i>
9	<i>Abnormal Update Rate</i>
10	<i>Abnormal Rate Of Change</i>
11	<i>Root Cause Not Known</i>
12	<i>Bad Component</i>
13	<i>Out Of Calibration</i>
14	<i>Special Instructions</i>
15	<i>Data Valid But Above Normal Operating Range – Least Severe Level</i>
16	<i>Data Valid But Above Normal Operating Range – Moderately Severe Level</i>
17	<i>Data Valid But Below Normal Operating Range – Least Severe Level</i>
18	<i>Data Valid But Below Normal Operating Range – Moderately Severe Level</i>
19	<i>Network Error</i>
20	<i>Data Drifted High</i>
21	<i>Data Drifted Low</i>
31	<i>Condition Exists</i>

**Table 20 – FMI for Event Options**

Every fault has associated a default FMI with them. The used FMI can be configured with **“FMI for Event”** setpoint, presented in Table 20. When an FMI is selected from Low Fault FMIs in Table 11

for a fault that can be flagged either high or low occurrence, it is recommended that the user would select the high occurrence FMI from the right column of Table 11. There is no automatic setting of High and Low FMIs in the firmware, the user can configure these freely.

<b>Low Fault FMIs</b>	<b>High Fault FMIs</b>
FMI=1, Data Valid But Below Normal Operation 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 Level – 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 11 – Low Fault FMIs and corresponding High Fault FMIs**

## 1.9. Simple Conditional Logic Function Blocks

The Simple Conditional Logic Blocks provide a way of connecting different blocks and comparing them to one another. The output of the Conditional Logic Blocks is either 0 (FALSE) or 1 (TRUE).

There are three operations that take place in the Conditional Logic Blocks which allows the user to make more elaborate comparisons. The first two operations, '**Operator 1 (Input 1, Input 2)**' and '**Operator 2 (Input 3, Input4)**' compare two separate values from different selectable control sources. The operations for these two operators are listed in Table 22.

Value	Operation	Meaning
0	<i>Not Used</i>	<i>Result = False</i>
1	=	<i>True when InA Equals InB</i>
2	!=	<i>True when InA Not Equal InB</i>
3	>	<i>True when InA Greater Than InB</i>
4	>=	<i>True when InA Greater Than or Equal InB</i>
5	<	<i>True when InA Less Than InB</i>
6	<=	<i>True when InA Less Than or Equal InB</i>
7	OR	<i>True when InA or InB is True</i>
8	AND	<i>True when InA and InB are True</i>
9	XOR	<i>True when InA/InB is True, but not both</i>

Table 22: Available operations for Operator 1 and Operator 2

The last operation is used between the results of both Operator 1 and Operator 2 using logic gates operations. The list of available operations for '**Operator 3 (Operator 1, Operator 2)**' are listed in Table 23.

Value	Operation	Meaning
0	<i>Not Used</i>	<i>Result = False</i>
1	OR	<i>True when Op1 or Op2 is True</i>
2	AND	<i>True when Op1 and Op2 are True</i>
3	XOR	<i>True when Op1/Op2 is True, but not both</i>

Table 23: Available operations for Operator 3



When configuring a Conditional Block and only one comparison is needed, it is necessary that operation selected for '**Operator 3 (Operator 1, Operator 2)**' is set to 1 ('OR, True When Op1 or Op2 is True')

## 1.10. Set / Reset Function Block

**Set-Reset Block** consists of only 2 control sources: **Reset Source** and **Set Source**. The purpose of these blocks is to simulate a modified latching function in which the 'Reset Signal' has more precedence. The 'latching' function works as per the Table 24 below.

'Set Signal'	'Reset Signal'	'Set-Reset Block Output' (Initial State: OFF)
OFF	OFF	<i>Latched State</i>
OFF	ON	OFF
ON	OFF	ON
ON	ON	OFF

**Table 24 – Set-Reset Function block operation**

The **Reset** and **Set** sources have associated with them a minimum and maximum threshold values which determine the ON and OFF state. For the **Reset Source** are **Reset Minimum Threshold** and **Reset Maximum Threshold**. Similarly for the **Set Source** are **Set Minimum Threshold** and **Set Maximum Threshold**. These setpoints also allow to have a deadband in between ON/OFF states and they are in terms of percentage of input selected.

As seen in Table 24 above, the 'Reset Signal' has more precedence over the 'Set Signal' - as long as the state of 'Reset Signal' is *ON*, the state of 'Set-Reset Block Output' will be *OFF*. In order to create an *ON* state in 'Set-Reset Block Output' the state of 'Reset Signal' must be *OFF* while the state of 'Set Signal' is *ON*. In this case, the state of 'Set-Reset Block Output' will remain *ON* even if 'Set Signal' turns *OFF* as long as 'Reset Signal' remains *OFF*. As soon as the 'Reset Signal' turns *ON* the 'Set-Reset Block Output' will turn *OFF* regardless of the state of 'Set Signal'.

## 1.11. Lookup Table Function Block

Lookup Tables are used to give an output response of up to 10 slopes per Lookup Table. There are two types of Lookup Table response based on **X-Axis Type**: *Data Response* and *Time Response* Sections 1.11.2 through 1.11.6 will describe these two **X-Axis Types** in more detail. If more than 10 slopes are required, a Programmable Logic Block can be used to combine up to three tables to get 30 slopes, as is described in Section 1.12.

There are two key setpoints that will affect this function block. The first is the **X-Axis Source** and **X-Axis Number** which together define the Control Source for the function block. When it is changed, the table is automatically updated with new defaults based on the X-Axis source selected if **Auto Update on Setpoint Changes** in the **Miscellaneous** block is *TRUE*.

### 1.11.1. Auto Update on Setpoint Changes

If **Auto Update on Setpoint Changes** is set to *TRUE*, the **X-Values** of the Lookup Tables are updated based on the **X-Axis Source** and **X-Axis Number** (i.e. the Min or Max values of the function block are updated), the associated table will also be automatically updated with default settings, based on the new X-Axis limits. However, if **Auto Update on Setpoint Changes** is set to *FALSE* then the **X-Values** will not get automatically updated on a setpoint change, but the minimum and maximum allowable ranges will be adjusted which the Axiomatic Electronic Assistant may give warnings for out-of-range setpoints. For example, if **Universal Input 1** is configured with **Minimum Range: 0.5V (Xmin)** and **Maximum Range: 9.8V (Xmax)** and is the **X-Axis Source** to **Lookup Table 1** with maximum **X Value** of 9.8V (*Xmax*), if **Universal Input 1** is then changed to **Maximum Range: 5.5V**, **Lookup Table 1** will not update its **X-Values** however, the maximum command (*Xmax*) will be set to 5.5V so editing the maximum **X Value** to something higher than 5.5V the Axiomatic Electronic Assistant will display a warning. It is up to the user's discretion to appropriately select the values in the Lookup Tables when **Auto Update on Setpoint Changes** is set to *FALSE*.

### 1.11.2. X-Axis, Input Data Response

In the case where the **X-Axis Type** = *Data Response*, the points on the X-Axis represents the data of the control source. These values must be selected within the range of the control source.

When selecting X-Axis data values, there are no constraints on the value that can be entered into any of the X-Axis points. The user should enter values in increasing order to be able to utilize the entire table. Therefore, when adjusting the X-Axis data, it is recommended that X<sub>10</sub> is changed first, then lower indexes in descending order as to maintain the below:

$$Xmin \leq X_0 \leq X_1 \leq X_2 \leq X_3 \leq X_4 \leq X_5 \leq X_6 \leq X_7 \leq X_8 \leq X_9 \leq X_{10} \leq Xmax$$

As stated earlier, *Xmin* and *Xmax* will be determined by the X-Axis Source that has been selected.

If some of the data points are '*Ignored*' as described in Section 1.11.4, they will not be used in the X-Axis calculation shown above. For example, if points X<sub>4</sub> and higher are ignored, the formula becomes *Xmin* ≤ X<sub>0</sub> ≤ X<sub>1</sub> ≤ X<sub>2</sub> ≤ X<sub>3</sub> ≤ *Xmax* instead.

### 1.11.3. Y-Axis, Lookup Table Output

The Y-Axis has no constraints on the data that it represents. This means that inverse, or increasing/decreasing or other responses can be easily established.

In all cases, the controller looks at the **entire range** of the data in the Y-Axis setpoints, and selects the lowest value as the *Ymin* and the highest value as the *Ymax*. They are passed directly to other function blocks as the limits on the Lookup Table output. (i.e used as Xmin and Xmax values in linear calculations.)

However, if some of the data points are '*Ignored*' as described in Section 1.11.4, they will not be used in the Y-Axis range determination. Only the Y-Axis values shown on the Axiomatic EA will be considered when establishing the limits of the table when it is used to drive another function block, such as a Math Function Block.

#### **1.11.4. Default Configuration, Data Response**

By default, all Lookup Tables in the ECU are disabled (**X-Axis Source** equals *Control Not Used*). Lookup Tables can be used to create the desired response profiles. If a Universal Input is used as the X-Axis, the output of the Lookup Table will be what the user enters in **Y-Values** setpoints.

Recall, any controlled function block which uses the Lookup Table as an input source will also apply a linearization to the data. **Therefore, for a 1:1 control response, ensure that the minimum and maximum values of the output correspond to the minimum and maximum values of the table's Y-Axis.**

All tables (1 to 3) are disabled by default (no control source selected). However, should an **X-Axis Source** be selected, the **Y-Values** defaults will be in the range of 0 to 100% as described in the "Y-Axis, Lookup Table Output" section above. X-Axis minimum and maximum defaults will be set as described in the "X-Axis, Data Response" section above.

**By default, the X and Y axes data is setup for an equal value between each point from the minimum to maximum in each case.**

#### **1.11.5. Point To Point Response**

By default, the X and Y axes are setup for a linear response from point (0,0) to (10,10), where the output will use linearization between each point, as shown in Figure 5. To get the linearization, each "**Point N – Response**", where N = 1 to 10, is setup for a '*Ramp To*' output response.

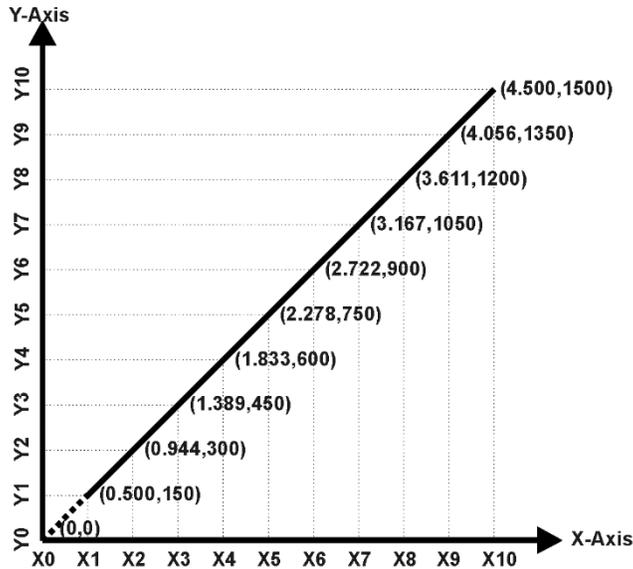


Figure 5 – Lookup Table with “Ramp To” Data Response

Alternatively, the user could select a ‘Jump To’ response for “Point N – Response”, where N = 1 to 10. In this case, any input value between  $X_{N-1}$  to  $X_N$  will result in an output from the Lookup Table function block of  $Y_N$ .

An example of a Math function block (0 to 100) used to control a default table (0 to 100) but with a ‘Jump To’ response instead of the default ‘Ramp To’ is shown in Figure 6.

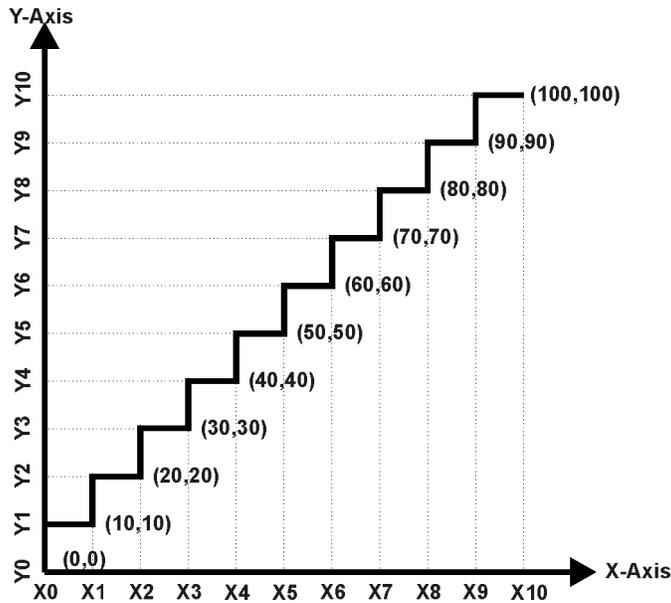


Figure 6 – Lookup Table with “Jump To” Data Response

Lastly, any point except (0,0) can be selected for an ‘Ignore’ response. If “Point N – Response” is set to ignore, then all points from  $(X_N, Y_N)$  to  $(X_{10}, Y_{10})$  will also be ignored. For all data greater than  $X_{N-1}$ , the output from the Lookup Table function block will be  $Y_{N-1}$ .

A combination of *Ramp To*, *Jump To* and *Ignore* responses can be used to create an application specific output profile.

### 1.11.6. X-Axis, Time Response

As mentioned in Section 1.11, a Lookup Table can also be used to get a custom output response where the **X-Axis Type** is a '*Time Response*.' When this is selected, the X-Axis now represents time, in units of milliseconds, while the Y-Axis still represents the output of the function block. There is also another setpoint associated to the Lookup Table when configured to *Time Response* which is the **Table Auto-Cycle** setpoint.

In this case, the **X-Axis Source** is treated as a digital input. If the signal is actually an analog input, it is interpreted like a digital input per Figure 2. When the control input is ON, the output will be changed over a period of time based on the profile in the Lookup Table. There are two different scenarios on how the Lookup Table will react once the profile is finished. The first option is when **Table Auto-Cycle** is set to *FALSE* in which case, once the profile has finished (i.e. index 10, or *Ignored* response), the output will remain at the last output at the end of the profile until the control input turns OFF. The second option is when **Table Auto-Cycle** is set to *TRUE* in which case, once the profile has finished (i.e. index 10, or *Ignored* response), the Lookup Table will automatically return to the 1<sup>st</sup> response and will continually be auto-cycling for as long as the input remains in the ON state.

When the control input is OFF, the output is always at zero. When the input comes ON, the profile ALWAYS starts at position (X<sub>0</sub>, Y<sub>0</sub>) which is 0 output for 0ms.

In a time response, the interval time between each point on the X-axis can be set anywhere from 1ms to 1min. [60,000 ms]

## 1.12. Programmable Logic Function Block

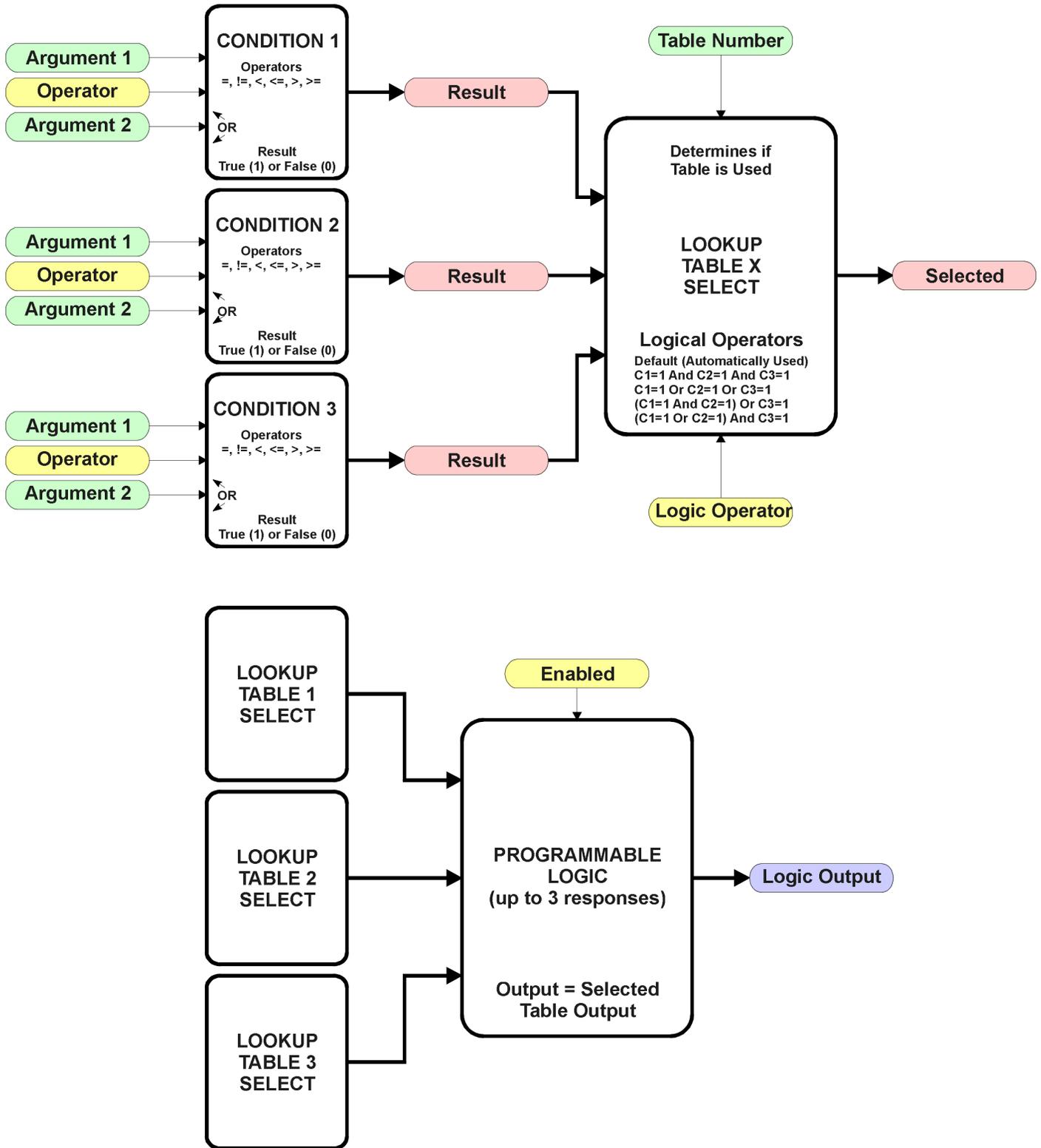


Figure 7 – Programmable Logic Function Block

This function block is obviously the most complicated of them all, but very powerful. The Programmable Logic can be linked to up to three tables, any one of which would be selected only under given conditions. Any three tables (of the available 8) can be associated with the logic, and which ones are used is fully configurable.

Should the conditions be such that a particular table (1, 2 or 3) has been selected as described in Section 1.12.2, then the output from the selected table, at any given time, will be passed directly to the Logic Output.

Therefore, up to three different responses to the same input, or three different responses to different inputs, can become the input to another function block, such as an Output X Drive. To do this, the “**Control Source**” for the reactive block would be selected to be the *‘Programmable Logic Function Block.’*

In order to enable any one of Programmable Logic blocks, the “**Programmable Logic Block Enabled**” setpoint must be set to True. They are all disabled by default.

Logic is evaluated in the order shown in Figure 8. Only if a lower number table has not been selected will the conditions for the next table be looked at. **The default table is always selected as soon as it is evaluated. It is therefore required that the default table always be the highest number in any configuration.**

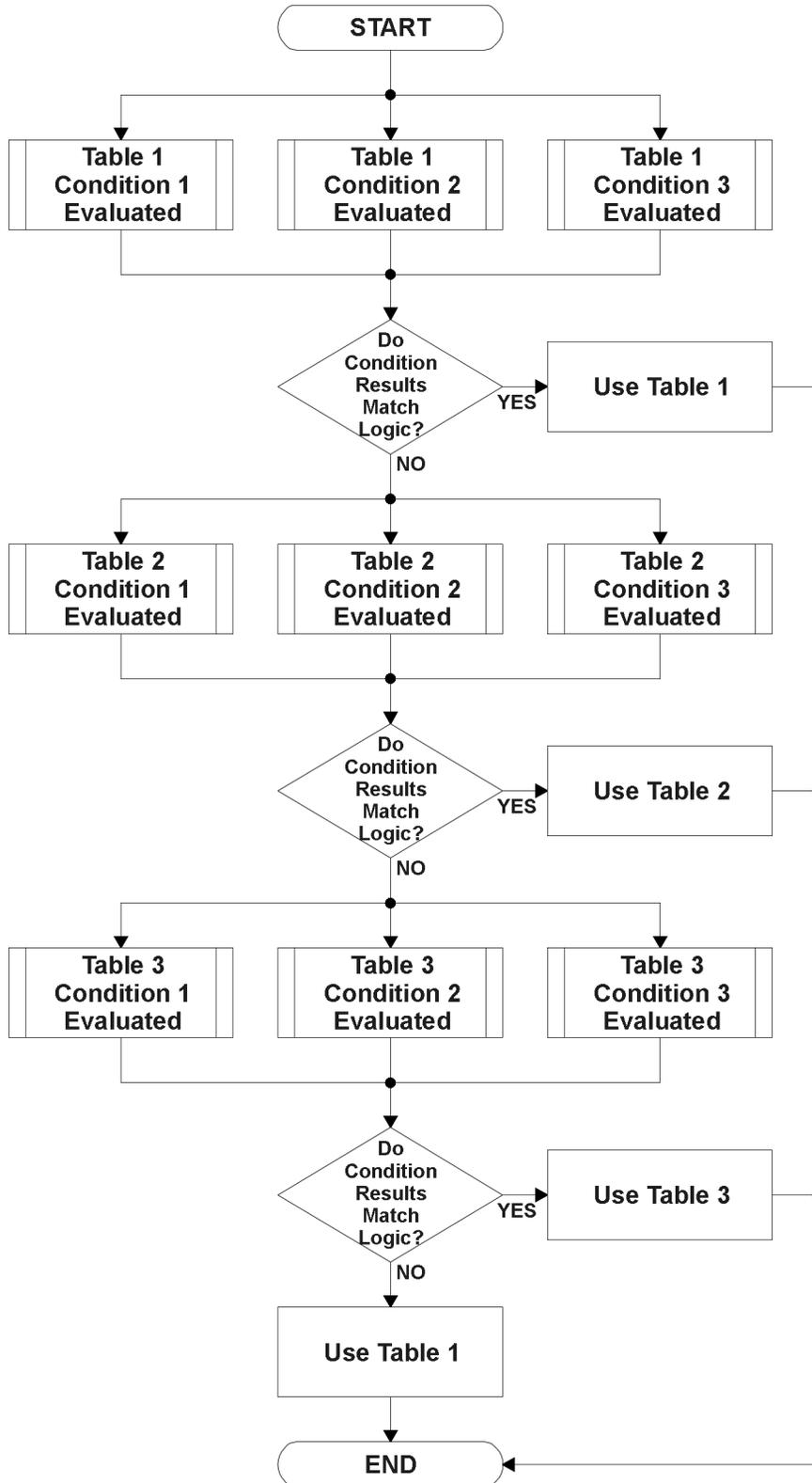


Figure 8 – Programmable Logic Flowchart

### 1.12.1. Conditions Evaluation

The first step in determining which table will be selected as the active table is to first evaluate the conditions associated with a given table. Each table has associated with it up to three conditions that can be evaluated.

Argument 1 is always a logical output from another function block, as listed in Section 1.4. As always, the source is a combination of the functional block type and number, setpoints “**Table X, Condition Y, Argument 1 Source**” and “**Table X, Condition Y, Argument 1 Number**”, where both X = 1 to 3 and Y = 1 to 3.

Argument 2 on the other hand, could either be another logical output such as with Argument 1, OR a constant value set by the user. To use a constant as the second argument in the operation, set “**Table X, Condition Y, Argument 2 Source**” to ‘*Control Constant Data.*’ Note that the constant value has no unit associated with it in the Axiomatic EA, so the user must set it as needed for the application.

The condition is evaluated based on the “**Table X, Condition Y Operator**” selected by the user. It is always ‘=, *Equal*’ by default. The only way to change this is to have two valid arguments selected for any given condition. Options for the operator are listed in Table 25.

0	=, <i>Equal</i>
1	!=, <i>Not Equal</i>
2	>, <i>Greater Than</i>
3	>=, <i>Greater Than or Equal</i>
4	<, <i>Less Than</i>
5	<=, <i>Less Than or Equal</i>

**Table 25 – Condition Operator Options**

For example, a condition for a weight going over a certain range could be that the Strain Gauge Input Measured 1 be greater than a certain value to flag a different output. In this case, “**...Argument 1 Source**” would be set to ‘*Strain Gauge Input Measured 1*, “**...Argument 2 Source**” to ‘*Control Constant Data*’, and the “**...Operator**” to ‘>, *Greater Than.*’ The “**Constant Value X**” in the Constant Data List would be set to whatever warning weight the application required.

By default, both arguments are set to ‘*Control Source Not Used*’ which disables the condition, and automatically results in a value of N/A as the result. Although Figure 7 shows only True or False as a result of a condition evaluation, the reality is that there could be four possible results, as described in Table 26.

Value	Meaning	Reason
0	False	(Argument 1) Operator (Argument 2) = False
1	True	(Argument 1) Operator (Argument 2) = True
2	Error	Argument 1 or 2 output was reported as being in an error state
3	Not Applicable	Argument 1 or 2 is not available (i.e. set to ‘ <i>Control Source Not Used</i> ’)

**Table 26– Condition Evaluation Results**

### 1.12.2. Table Selection

In order to determine if a particular table will be selected, logical operations are performed on the results of the conditions as determined by the logic in Section 1.12.1. There are several logical combinations that can be selected, as listed in Table 27.

0	<i>Default Table</i>
1	<i>Cnd1 And Cnd2 And Cnd3</i>
2	<i>Cnd1 Or Cnd2 Or Cnd3</i>
3	<i>(Cnd1 And Cnd2) Or Cnd3</i>
4	<i>(Cnd1 Or Cnd2) And Cnd3</i>

**Table 27 – Conditions Logical Operator Options**

Not every evaluation is going to need all three conditions. The case given in the earlier section, for example, only has one condition listed, i.e. that the Engine RPM be below a certain value. Therefore, it is important to understand how the logical operators would evaluate an Error or N/A result for a condition.

<b>Logical Operator</b>	<b>Select Conditions Criteria</b>
Default Table	Associated table is automatically selected as soon as it is evaluated.
Cnd1 And Cnd2 And Cnd3	<p><b>Should be used when two or three conditions are relevant, and all must be true to select the table.</b></p> <p>If any condition equals False or Error, the table is not selected. An N/A is treated like a True. If all three conditions are True (or N/A), the table is selected.</p> <p>If((Cnd1==True) &amp;&amp;(Cnd2==True)&amp;&amp;(Cnd3==True)) Then Use Table</p>
Cnd1 Or Cnd2 Or Cnd3	<p><b>Should be used when only one condition is relevant. Can also be used with two or three relevant conditions.</b></p> <p>If any condition is evaluated as True, the table is selected. Error or N/A results are treated as False</p> <p>If((Cnd1==True)    (Cnd2==True)    (Cnd3==True)) Then Use Table</p>
(Cnd1 And Cnd2) Or Cnd3	<p><b>To be used only when all three conditions are relevant.</b></p> <p>If both Condition 1 and Condition 2 are True, OR Condition 3 is True, the table is selected. Error or N/A results are treated as False</p> <p>If( ((Cnd1==True)&amp;&amp;(Cnd2==True))    (Cnd3==True) ) Then Use Table</p>
(Cnd1 Or Cnd2) And Cnd3	<p><b>To be used only when all three conditions are relevant.</b></p> <p>If Condition 1 And Condition 3 are True, OR Condition 2 And Condition 3 are True, the table is selected. Error or N/A results are treated as False</p> <p>If( ((Cnd1==True)  ((Cnd2==True)) &amp;&amp; (Cnd3==True) ) Then Use Table</p>

**Table 28 – Conditions Evaluation Based on Selected Logical Operator**

The default “**Table X, Conditions Logical Operator**” for Table 1 and Table 2 is ‘*Cnd1 And Cnd2 And Cnd3*,’ while Table 3 is set to be the ‘*Default Table*.’

### 1.12.3. Logic Block Output

Recall that Table X, where X = 1 to 3 in the Programmable Logic function block does NOT mean Lookup Table 1 to 3. Each table has a setpoint “**Table X – Lookup Table Block Number**” which allows the user to select which Lookup Tables they want associated with a particular Programmable Logic Block. The default tables associated with each logic block are listed in Table 17.

<b>Programmable Logic Block Number</b>	<b>Table 1 – Lookup Table Block Number</b>	<b>Table 2 – Lookup Table Block Number</b>	<b>Table 3 – Lookup Table Block Number</b>
1	1	2	3

Table 29 – Programmable Logic Block Default Lookup Tables

If the associated Lookup Table does not have an “**X-Axis Source**” selected, then the output of the Programmable Logic block will always be “Not Available” so long as that table is selected. However, should the Lookup Table be configured for a valid response to an input, be it Data or Time, the output of the Lookup Table function block (i.e. the Y-Axis data that has been selected based on the X-Axis value) will become the output of the Programmable Logic function block so long as that table is selected.

Unlike all other function blocks, the Programmable Logic does NOT perform any linearization calculations between the input and the output data. Instead, it mirrors exactly the input (Lookup Table) data. Therefore, when using the Programmable Logic as a control source for another function block, it is HIGHLY recommended that all the associated Lookup Table Y-Axes either be (a) Set between the 0 to 100% output range or (b) all set to the same scale.

### 1.13. Math Function Block

There are four mathematical function blocks that allow the user to define basic algorithms. A math function block can take up to four input signals. Each input is then scaled according to the associated limit and scaling setpoints.

Inputs are converted into percentage value based on the “**Function X Input Y Minimum**” and “**Function X Input Y Maximum**” values selected. For additional control the user can also adjust the “**Function X Input Y Scaler**”. By default, each input has a scaling ‘weight’ of 1.0 However, each input can be scaled from -1.0 to 1.0 as necessary before it is applied in the function.

A mathematical function block includes three selectable functions, which each implements equation  $A \text{ operator } B$ , where A and B are function inputs and operator is function selected with setpoint **Math function X Operator**. Setpoint options are presented in Table 30. The functions are connected together, so that result of the preceding function goes into Input A of the next function. Thus Function 1 has both Input A and Input B selectable with setpoints, where Functions 2 to 4 have only Input B selectable. Input is selected by setting **Function X Input Y Source** and **Function X Input Y Number**. If **Function X Input B Source** is set to 0 *Control not used* signal goes through function unchanged.

$$\text{Math Block Output} = (((A1 \text{ op1 } B1) \text{ op2 } B2) \text{ op3 } B3) \text{ op4 } B4$$

0	=, True when InA equals InB
1	!=, True when InA not equal InB
2	>, True when InA greater than InB
3	>=, True when InA greater than or equal InB
4	<, True when InA less than InB
5	<=, True when InA less than or equal InB
6	OR, True when InA or InB is True
7	AND, True when InA and InB are True
8	XOR, True when either InA or InB is True, but not both
9	+, Result = InA plus InB
10	-, Result = InA minus InB
11	x, Result = InA times InB
12	/, Result = InA divided by InB
13	MIN, Result = Smallest of InA and InB
14	MAX, Result = Largest of InA and InB
15	MAX-MIN, Result = Absolute value of (InA – InB)

**Table 30 – Math Function Operators**

User should make sure the inputs are compatible with each other when using some of the Mathematical Operations. For instance, if Universal Input 1 is to be measured in [V], while CAN Receive 1 is to be measured in [mV] and Math Function Operator 9 (+), the result will not be the true value desired.

For a valid result, the control source for an input must be a non-zero value, i.e. something other than ‘Control Source Not Used.’

When dividing, a zero InB value will always result in a zero output value for the associated function. When subtracting, a negative result will always be treated as a zero, unless the function is multiplied by a negative one, or the inputs are scaled with a negative coefficient first.

#### 1.14. 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. 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 5 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 1IN-1OUT-LED are all for proprietary SPNs, and are defined in detail in Section 3.3.

Note:  $CAN\ Data = (Input\ Data - Offset) / Resolution$

The 1IN-1OUT-LED supports up to 5 unique CAN Transmit Messages, all of which can be programmed to send any available data to the CAN network.

## 1.15. Diagnostic Trouble Code (DTC) React

The DTC React function block will allow a received DTC sent from another ECU on a DM1 message to be used as an input to any other function block in order to disable an output, for example. Up to three SPN/FMI combinations can be selected.

Should a DM1 message be received with the SPN/FMI combination defined, the corresponding DTC State will be set to ON. Once ON, if the same SPN/FMI combination has not been received again after 3 seconds, the DTC State will be reset to OFF.

The DTC could be used as a digital input for any function block as appropriate.

## 1.16. 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 another function block.

The **Receive 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 receive messages are disabled.

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

By default, all control messages are expected to be sent to the 1IN-1OUT-LED Controller on Proprietary B PGNs. However, should a PDU1 message be selected, the 1IN-1OUT-LED Controller 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 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 used to map any SPN supported by the J1939 standard to the output data of the Received function block.

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 determine the minimum and maximum values of the control signal. As the names imply, they are also used as the On/Off thresholds for digital output types. These values are in whatever units the data is AFTER the resolution and offset is applied to CAN receive signal. The 1IN-1OUT-LED Controller supports up to five unique CAN Receive Messages.

## 2. Installation Instructions

### 2.1. Dimensions and Pinout

The 1IN-1OUT-LED Controller is packaged in an ultra-sonically welded plastic housing. The assembly carries an IP67 rating.

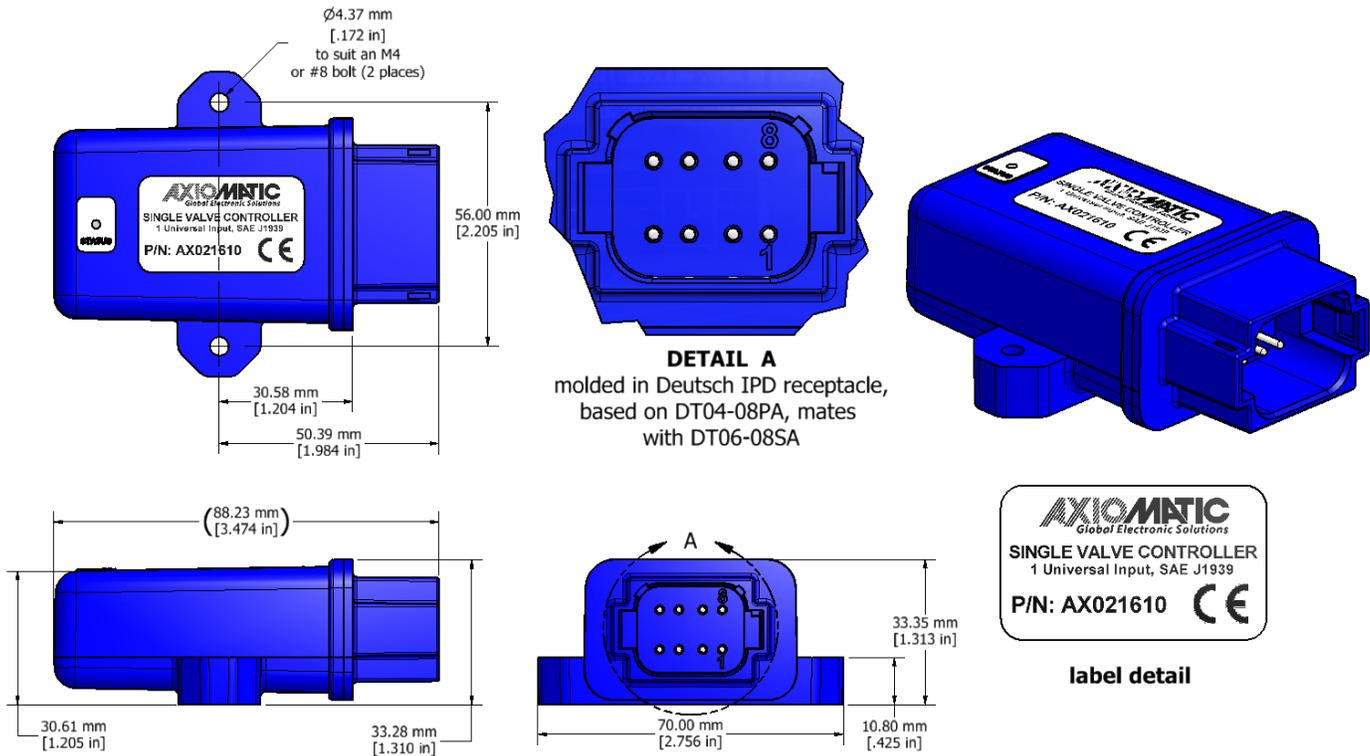


Figure 9 – Housing Dimensions

PIN #	FUNCTION
1	CAN_L
2	CAN_H
3	Output GND
4	Universal Input
5	Input Analog GND
6	Output +
7	Batt -
8	Batt +

Table 31 – Connector Pinout

### 2.2. Mounting Instructions

#### NOTES & WARNINGS

- Do not install near high-voltage or high-current devices.
- Note the operating temperature range. All field wiring must be suitable for that temperature range.
- Install the unit with appropriate space available for servicing and for adequate wire harness access (15 cm) and strain relief (30 cm).
- Do not connect or disconnect the unit while the circuit is live, unless the area is known to be non-hazardous.

## MOUNTING

Mounting holes are sized for #8 or M4 bolts. The bolt length will be determined by the end-user's mounting plate thickness. The mounting flange of the controller is 0.425 inches (10.8 mm) thick.

If the module is mounted without an enclosure, it should be mounted vertically with connectors facing left or right to reduce likelihood of moisture entry.

The CAN wiring is considered intrinsically safe. The power wires are not considered intrinsically safe and so in hazardous locations they need to be located in conduit or conduit trays at all times. The module must be mounted in an enclosure in hazardous locations for this purpose.

No wire or cable harness should exceed 30 meters in length. The power input wiring should be limited to 10 meters.

All field wiring should be suitable for the operating temperature range.

Install the unit with appropriate space available for servicing and for adequate wire harness access (6 inches or 15 cm) and strain relief (12 inches or 30 cm).

## CONNECTIONS

Use the following TE Deutsch mating plugs to connect to the integral receptacles. Wiring to these mating plugs must be in accordance with all applicable local codes. Suitable field wiring for the rated voltage and current must be used. The rating of the connecting cables must be at least 85°C. For ambient temperatures below -10°C and above +70°C, use field wiring suitable for both minimum and maximum ambient temperature.

Refer to the respective TE Deutsch datasheets for usable insulation diameter ranges and other instructions.

Receptacle Contacts	Mating Sockets as appropriate (Refer to <a href="http://www.laddinc.com">www.laddinc.com</a> for more information on the contacts available for this mating plug.)
Mating Connector	DT06-08SA, 1 W8S, 8 0462-201-16141, and 3 114017

### 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 Transmit PGN and SPN Parameters
- Configurable Receive PGN and SPN Parameters
- Sending DM1 Diagnostic Message Parameters
- Reading and reacting to DM1 messages sent by other ECUs
- 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) |

Note: Any Proprietary B PGN in the range 65280 to 65535 (\$00FF00 to \$00FFFF) can be selected

##### 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 are 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

#### J1939 NAME

The 1IN-1OUT-LED 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	125, Axiomatic I/O Controller
Function Instance	16, Axiomatic AX021610, Single Input Single Valve Output Controller
<b>ECU Instance</b>	<b>0, First Instance</b>
Manufacture Code	162, Axiomatic Technologies Corporation
Identity Number	Variable, uniquely assigned during factory programming for each ECU

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 128 (0x80), which is the preferred starting address for self-configurable ECUs 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 1IN-1OUT-LED will continue select the next highest address until it find one that it can claim. See J1939/81 for more details about address claiming.

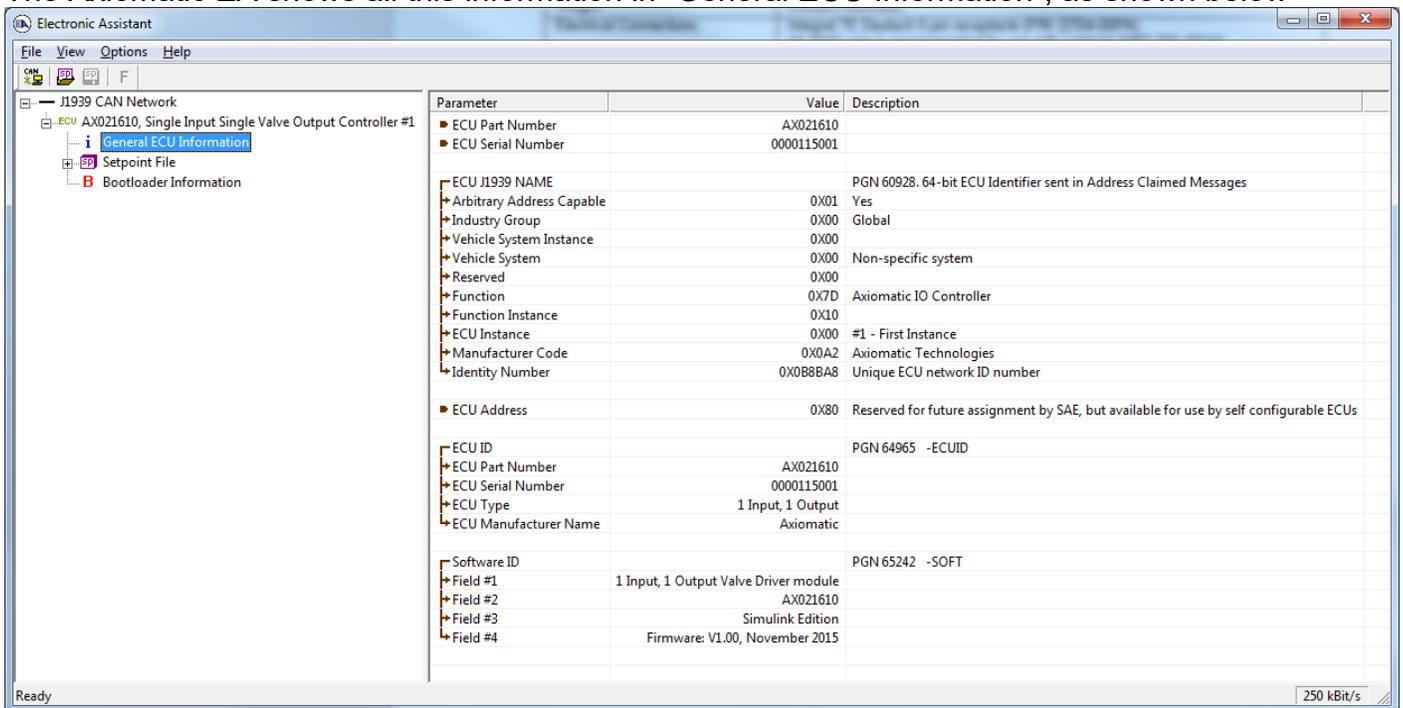
#### Software Identifier

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 1IN-1OUT-LED 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



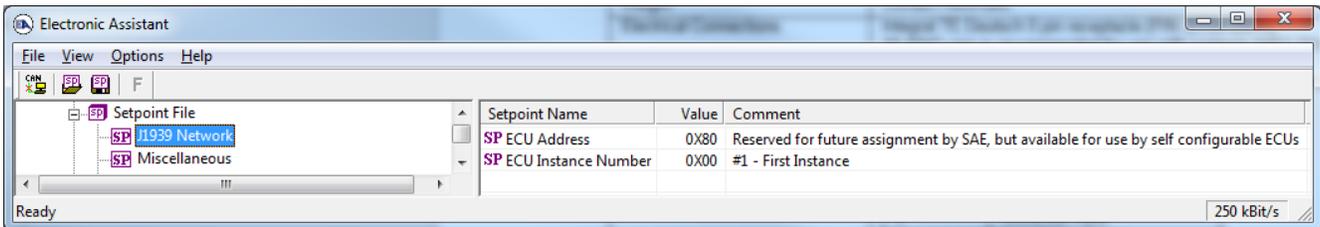
*Note: The information provided in the Software ID is available for any J1939 service tool which supports the PGN -SOFT.*

## 4. ECU SETPOINTS ACCESSED WITH THE AXIOMATIC ELECTRONIC ASSISTANT

Many setpoints have been reference 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 1IN-1OUT-LED, refer to the relevant section of the User Manual.

### 4.1. J1939 Network

The J1939 Network setpoints deal with the controller’s parameters specifically affecting the CAN network. Refer to the notes on information about each setpoint.



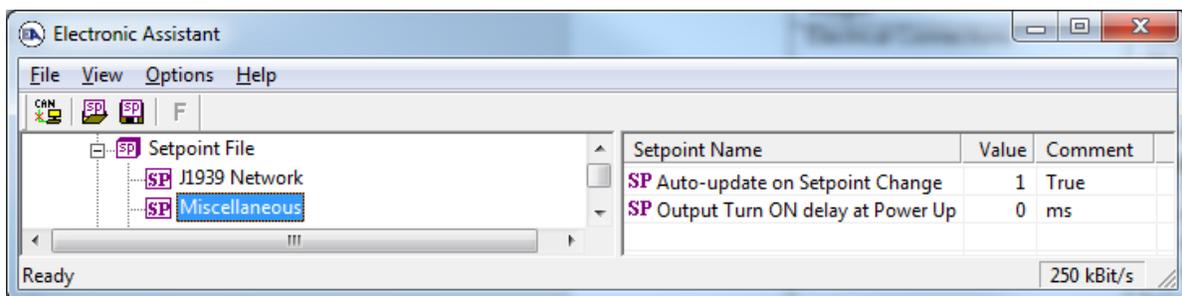
Name	Range	Default	Notes
ECU Instance Number	Drop List	0, #1 – First Instance	Per J1939-81
ECU Address	0 to 253	128 (0x80)	Preferred address for a self-configurable ECU

*Screen Capture of Default Miscellaneous Setpoints*

If non-default values for the “**ECU Instance Number**” or “**ECU Address**” are used, they will not be updated during a setpoint file flash. These parameters need to be changed manually in order to prevent other units on the network to be affected. When they are changed, the controller will claim its new address on the network. It is recommended to close and re-open the CAN connection on the Axiomatic EA after the file is loaded, such that only the new NAME and address appear in the J1939 CAN Network ECU list.

### 4.2. Miscellaneous Setpoints

The Miscellaneous setpoints primarily deal with the CAN Network. Refer to the notes for more information about each setpoint.



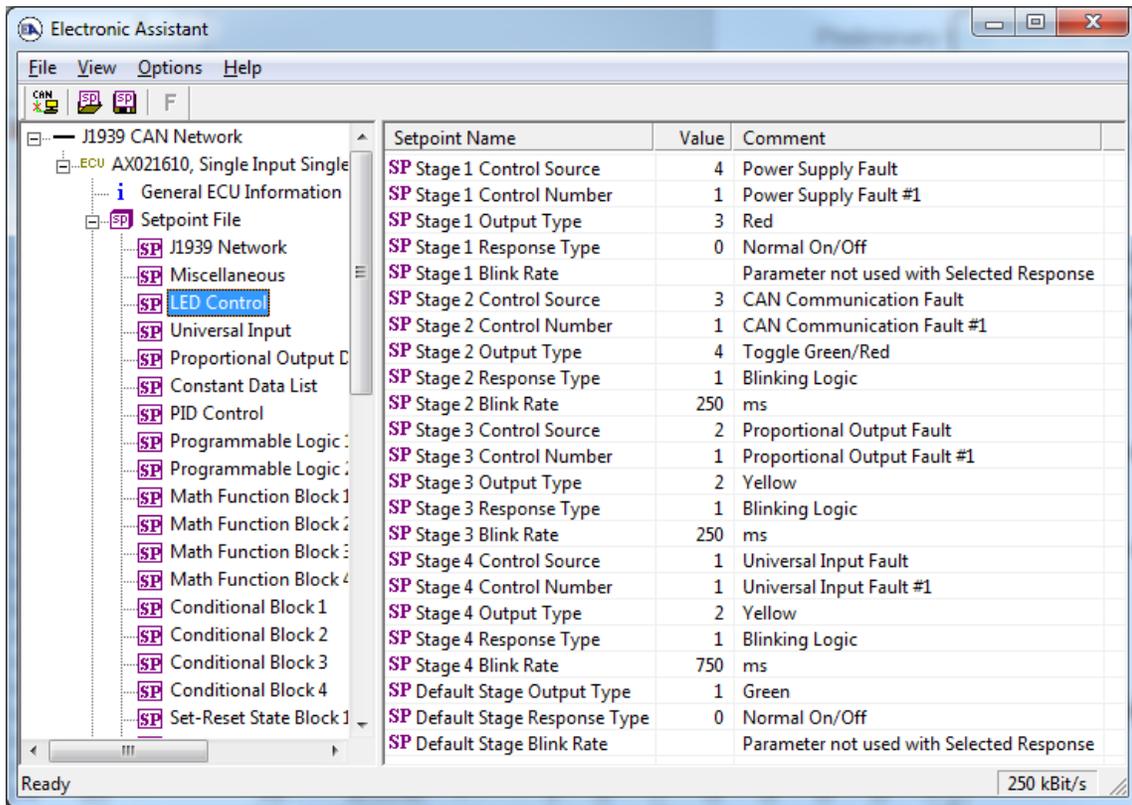
*Screen Capture of Default Miscellaneous Setpoints*

Name	Range	Default	Notes
Auto-update on Setpoint Change	Drop List	True	If set to <i>TRUE</i> , the Lookup tables get updated automatically when the control source is updated. If set to <i>FALSE</i> , parameters remain after control source has been updated
Output Turn ON delay at Power Up	0 to 60,000 [ms]	0 [ms]	This setpoint specifies the delay the output will undergo before going active at power up.

**Output Turn ON delay at Power Up** setpoint allows to have an output start-up delay before the output is commanded (if input signal exists).

### 4.3. LED Control

The Dual LED control function block is defined in Section 1.2. Please refer to that section for detailed information on how these setpoints are used.



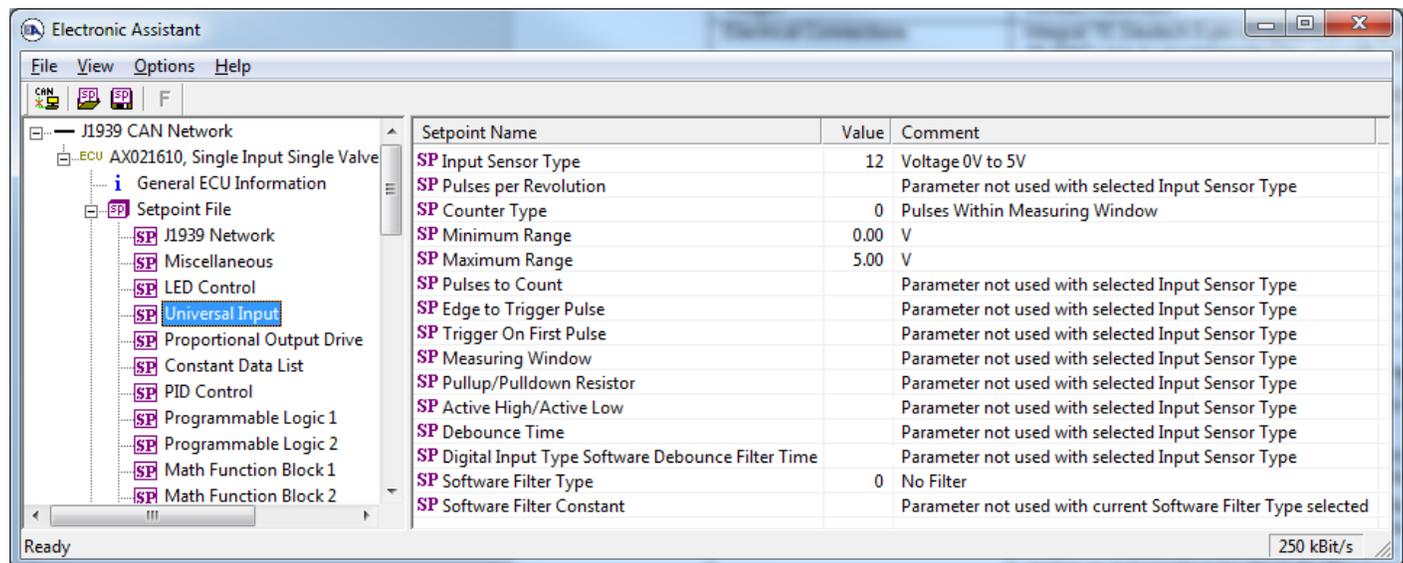
**Screen Capture of an Example for LED Control Function Block**

*Note: In the screen capture shown above, the setpoints have been changed from their default values in order to better illustrate their functionality when used. The priority is structured such that Stage 1 is highest priority and Stage 4 is lowest priority. If no stage is active, the default stage is used. The table below outlines the default values, while the screen capture above illustrates its configurability/useability.*

Name	Range	Default	Notes
Stage n Control Source	Drop List	0 – Control Not Used	Refer to Section 1.5
Stage n Control Number	Depends on Control Source	1	Range depends on Control Source selected
Stage n Output Type	Drop List	1 – Green	Refer to Section 1.2
Stage n Response Type	Drop List	1 – Blinking Logic	Refer to Section 1.2
Stage n Blink Rate	0 to 60000	1000 (ms)	
Default Stage Output Type	Drop List	1 – Green	
Default Stage Response Type	Drop List	1 – Blinking Logic	
Default Stage Blink Rate	0 to 60000	1000ms	
Where n = 1..4			

## 4.4. Universal Input

The Universal Input function block is defined in Section 1.3. Please refer to that section for detailed information on how these setpoints are used.



Screen Capture of Default Universal Input Setpoints

Name	Range	Default	Notes
Input Sensor Type	Drop List	12 – Voltage 0V to 5V	Refer to Section 1.3.1
Pulses per Revolution	0 to 60000	0	If set to 0, measurements are taken in Hz. If value is set greater than 0, measurements are taken in RPM
Counter Type	Drop List	0 – Pulses within Measuring Window	Refer to Section 1.3.4
Minimum Range	Depends on Sensor Type	0 (V)	
Maximum Range	Depends on Sensor Type	5 (V)	
Pulses to Count	0 to 10000	100 Pulses	
Edge to Trigger Pulse	Drop List	0 – <i>Falling Edge</i>	<i>Rising Edge or Falling Edge.</i> Edge capture which will trigger a pulse count increment.
Trigger on First Pulse	Drop List	0 - <i>False</i>	When <b>Pulses to Count</b> is set to >1, this setpoint determines to trigger on first pulse measured, or last.
Measuring Window	0 to 10000	1000 (ms)	Time to measure how many pulses are to be captured
Pullup/Pulldown Resistor	Drop List	0 – <i>Pullup/down Off</i>	Refer to Section 1.3.2
Active High/Active Low	Drop List	0 – <i>Active High</i>	Refer to Section 1.3.3
Debounce Time	Drop List	0 - <i>None</i>	
Digital Input Type Software Debounce Filter Type	0 to 60000	0 (ms)	Debounce time for Digital On/Off input type
Software Filter Type	Drop List	0 – <i>No Filter</i>	Refer to Section 1.3.6. This function is not used in Digital and Counter input types
Software Filter Constant	0 to 60000	1000ms	Refer to Section 1.3.6

## 4.5. Proportional Output Drive

The Universal Input function block is defined in Section 1.6. Please refer to that section for detailed information on how these setpoints are used.

Setpoint Name	Value	Comment
SP Output Type	1	Proportional Current
SP Output At Minimum Command	300	mA
SP Output At Maximum Command	1500	mA
SP Output At Override Command	750	mA
SP Dither Frequency	200	Hz
SP Dither Amplitude	0	mA
SP Ramp Up (Min to Max)	1000	ms
SP Ramp Down (Max to Mix)	1000	ms
SP PWM Output Frequency	25000	Hz
SP Hold Current		Parameter not used with current Output Type selected
SP Hotshot Current		Parameter not used with current Output Type selected
SP Hotshot Time		Parameter not used with current Output Type selected
SP Digital Response		Parameter not used with current Output Type selected
SP Digital Override State		Parameter not used with current Output Type selected
SP Digital Blink Rate		Parameter not used with current Output Type selected
SP Digital ON Time Control Source		Parameter not used with current Output Type selected
SP Digital ON Time Control Number		Parameter not used with current Output Type selected
SP Digital OFF Time Control Source		Parameter not used with current Output Type selected
SP Digital OFF Time Control Number		Parameter not used with current Output Type selected
SP Complete Full ON/OFF Cycle		Parameter not used with current Output Type selected
SP Control Source	2	Universal Input Measured
SP Control Number	1	Universal Input Measured #1
SP Enable Source	0	Control Not Used
SP Enable Number		Parameter not used with current Enable Source selected
SP Enable Response		Parameter not used with current Enable Source selected
SP Override Source	0	Control Not Used
SP Override Number		Parameter not used with current Override Source selected
SP Override Response		Parameter not used with current Override Source selected
SP Output Fault Response	0	Shutoff Output
SP Output in Fault Mode		Parameter not used with current Control Fault Response
SP Output Regulator Proportional Gain	0.6428100	
SP Output Regulator Integral Gain	0.0301200	
SP Output Regulator Derivative Gain	0.1421000	

**Screen Capture of Default Universal Input Setpoints**

Name	Range	Default	Notes
Output Type	Drop List	12 – Voltage 0V to 5V	Refer to Section 1.3.1
Output At Minimum Command	0 to 60000	0	If set to 0, measurements are taken in Hz. If value is set greater than 0, measurements are taken in RPM
Output At Maximum Command	Drop List	0 – Pulses within Measuring Window	Refer to Section 1.3.4
Output At Override Command	Depends on Sensor Type	0 (V)	
Dither Frequency	Depends on Sensor Type	5 (V)	
Dither Amplitude	0 to 10000	100 Pulses	
Ramp Up (Min to Max)	Drop List	0 – <i>Falling Edge</i>	Edge capture which will trigger an pulse count increment
Ramp Down (Max to Min)	Drop List	0 - <i>False</i>	When <b>Pulses to Count</b> is set to >1, this setpoint determines to trigger on first pulse measured, or last.
PWM Output Frequency	1 to 25000	25000 Hz	In PWM Output type, the frequency can be changed within the range specified. Other types will default the frequency to 25kHz
Hold Current	Drop List	0 – <i>Pullup/down Off</i>	Refer to Section 1.6
Hotshot Current	Drop List	0 – <i>Active High</i>	Refer to Section 1.6
Hotshot Time	Drop List	0 - <i>None</i>	Refer to Section 1.6
Digital Response	Drop List	0 – Normal On/Off	Refer to Section 1.6
Digital Blink Rate	0 to 60000	1000 ms	Used only with <i>Blinking Logic</i>
Digital ON Time Control Source	Drop List	0 – Not Used	Used only with <i>Period Logic</i> . Can be dynamic if source controlling the ON time varies. Fixed if using Constant Data List
Digital ON Time Control Number	Depends on Source	1	Refer to Section 1.4
Digital OFF Time Control Source	Drop List	0 – Not Used	Used only with <i>Period Logic</i> . Can be dynamic if source controlling the ON time varies. Fixed if using Constant Data List
Digital OFF Time Control Number	Depends on Source	1	Refer to Section 1.4
Complete Full ON/OFF Cycle	Drop List	0 – <i>FALSE</i>	<i>TRUE or FALSE</i> . Used only with <i>Period Logic</i>
Control Source	Drop List	2 – <i>Universal Input Measured</i>	Refer to Section 1.4
Control Number	Depends on Source	1	Refer to Section 1.4
Enable Source	Drop List	0 – <i>Control Not Used</i>	Refer to Section 1.4
Enable Number	Depends on Source	1	Refer to Section 1.4
Enable Response	Drop List	0 – <i>Enable When On, Else Shutoff</i>	Refer to Section 1.6
Override Source	Drop List	0 – <i>Control Not Used</i>	Refer to Section 1.4
Override Number	Depends on Source	1	Refer to Section 1.4
Override Response	Drop List	0 – <i>Override when ON</i>	Refer to Section 1.6
Output Fault Response	Drop List	0 – <i>Shutoff Output</i>	Refer to Section 1.6

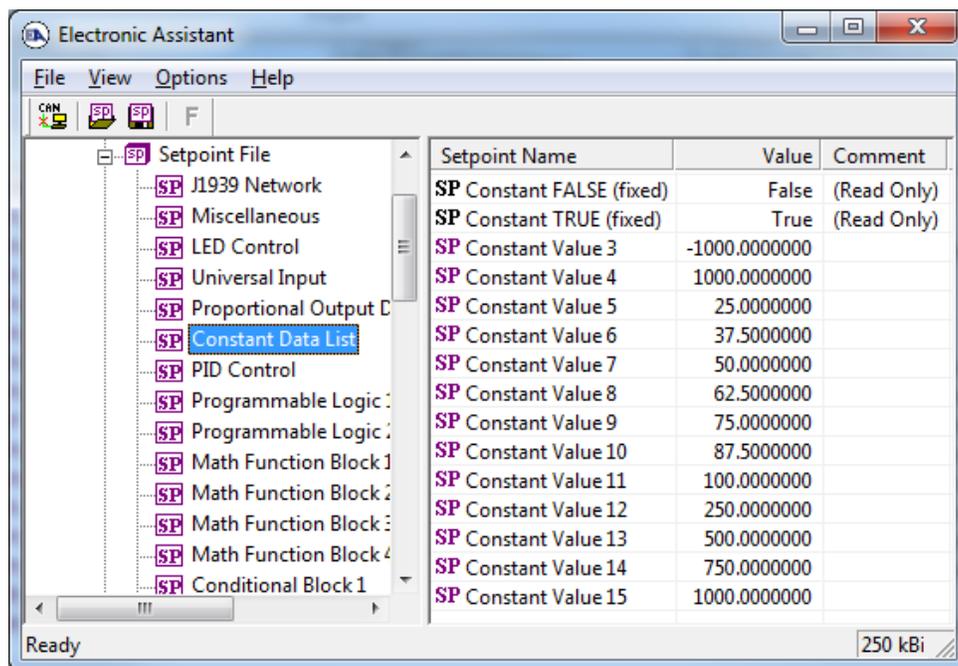
Output in Fault Mode	Depends on Type	500mA	Used only when <b>Output Fault Response</b> is <i>Apply Fault Value</i>
Output Regulator Proportional Gain	0 to 100	0.61281	Password Protected. Contact Axiomatic
Output Regulator Integral Gain	0 to 100	0.03012	Password Protected. Contact Axiomatic
Output Regulator Derivative Gain	0 to 100	0.1421	Password Protected. Contact Axiomatic

#### 4.6. Constant Data List Setpoints

The Constant Data List function block is provided to allow the user to select values as desired for various logic block functions. Throughout this manual, various references have been made to constants, as summarized in the examples listed below.

- a) Programmable Logic: Constant “**Table X = Condition Y, Argument 2**”, where X and Y = 1 to 3
- b) Math Function: Constant “**Math Input X**”, where X = 1 to 4

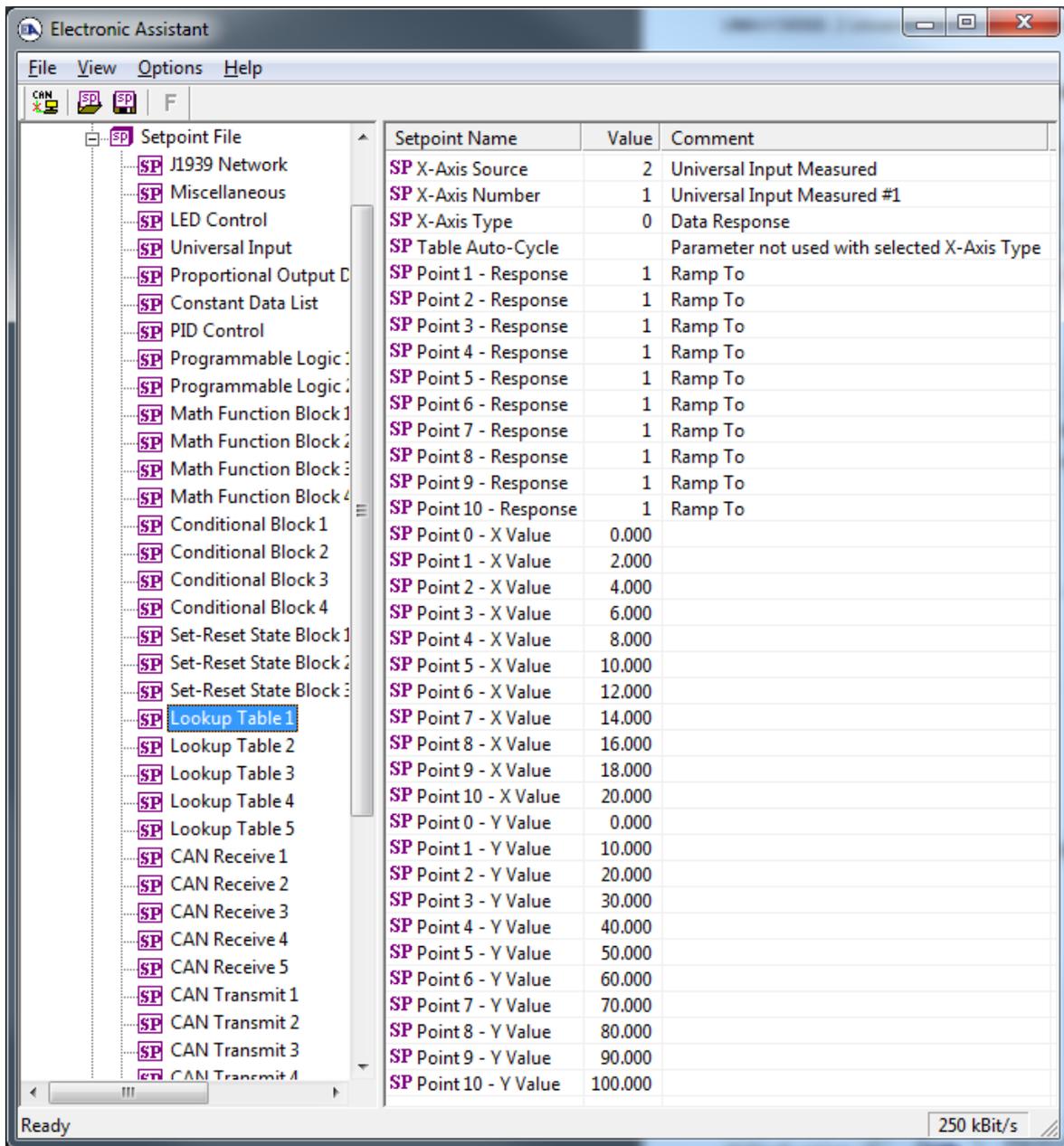
The first two constants are fixed values of 0 (False) and 1 (True) for use in binary logic. The remaining 13 constants are fully user configurable to any value between +/- 1,000,000. The default values are displayed in the screen capture below.



Screen Capture Default Constant Data List Setpoints

#### 4.7. Lookup Table Setpoints

The Lookup Table function block is defined in Section 1.11. Please refer there for detailed information about how all these setpoints are used. As this function block’s X-Axis defaults are defined by the “**X-Axis Source**” selected from Table 9, there is nothing further to define in terms of defaults and ranges beyond that which is described in Section 1.11. Recall, the X-Axis values will be automatically updated if the min/max range of the selected source is changed.



**Screen Capture of Example Lookup Table 1 Setpoints**

*Note: In the screen capture shown above, the “X-Axis Source” has been changed from its default value in order to enable the function block.*

#### **4.8. Programmable Logic Setpoints**

The Programmable Logic function block is defined in Section 1.12. Please refer there for detailed information about how all these setpoints are used.

As this function block is disabled by default, there is nothing further to define in terms of defaults and ranges beyond that which is described in Section 1.12. The screen capture below shows how the setpoints referenced in that section appear on the Axiomatic EA.

Setpoint Name	Value	Comment
SP Programmable Logic Enabled	1	True
SP Table 1 - Lookup Table Block Number	1	Lookup Table 1
SP Table 1 - Conditions Logical Operator	1	Cnd1 And Cnd2 And Cnd3
SP Table 1 - Condition 1, Argument 1 Source	0	Control Not Used
SP Table 1 - Condition 1, Argument 1 Number		Parameter not used with current Control Source selected
SP Table 1 - Condition 1, Operator		Parameter not used with current Control Source selected
SP Table 1 - Condition 1, Argument 2 Source	0	Control Not Used
SP Table 1 - Condition 1, Argument 2 Number		Parameter not used with current Control Source selected
SP Table 1 - Condition 2, Argument 1 Source	0	Control Not Used
SP Table 1 - Condition 2, Argument 1 Number		Parameter not used with current Control Source selected
SP Table 1 - Condition 2, Operator		Parameter not used with current Control Source selected
SP Table 1 - Condition 2, Argument 2 Source	0	Control Not Used
SP Table 1 - Condition 2, Argument 2 Number		Parameter not used with current Control Source selected
SP Table 1 - Condition 3, Argument 1 Source	0	Control Not Used
SP Table 1 - Condition 3, Argument 1 Number		Parameter not used with current Control Source selected
SP Table 1 - Condition 3, Operator		Parameter not used with current Control Source selected
SP Table 1 - Condition 3, Argument 2 Source	0	Control Not Used
SP Table 1 - Condition 3, Argument 2 Number		Parameter not used with current Control Source selected
SP Table 2 - Lookup Table Block Number	2	Lookup Table 2
SP Table 2 - Conditions Logical Operator	1	Cnd1 And Cnd2 And Cnd3
SP Table 2 - Condition 1, Argument 1 Source	0	Control Not Used
SP Table 2 - Condition 1, Argument 1 Number		Parameter not used with current Control Source selected
SP Table 2 - Condition 1, Operator		Parameter not used with current Control Source selected
SP Table 2 - Condition 1, Argument 2 Source	0	Control Not Used
SP Table 2 - Condition 1, Argument 2 Number		Parameter not used with current Control Source selected
SP Table 2 - Condition 2, Argument 1 Source	0	Control Not Used
SP Table 2 - Condition 2, Argument 1 Number		Parameter not used with current Control Source selected
SP Table 2 - Condition 2, Operator		Parameter not used with current Control Source selected
SP Table 2 - Condition 2, Argument 2 Source	0	Control Not Used
SP Table 2 - Condition 2, Argument 2 Number		Parameter not used with current Control Source selected
SP Table 2 - Condition 3, Argument 1 Source	0	Control Not Used
SP Table 2 - Condition 3, Argument 1 Number		Parameter not used with current Control Source selected
SP Table 2 - Condition 3, Operator		Parameter not used with current Control Source selected
SP Table 2 - Condition 3, Argument 2 Source	0	Control Not Used
SP Table 2 - Condition 3, Argument 2 Number		Parameter not used with current Control Source selected
SP Table 3 - Lookup Table Block Number	3	Lookup Table 3
SP Table 3 - Conditions Logical Operator	1	Cnd1 And Cnd2 And Cnd3
SP Table 3 - Condition 1, Argument 1 Source	0	Control Not Used
SP Table 3 - Condition 1, Argument 1 Number		Parameter not used with current Control Source selected
SP Table 3 - Condition 1, Operator		Parameter not used with current Control Source selected
SP Table 3 - Condition 1, Argument 2 Source	0	Control Not Used
SP Table 3 - Condition 1, Argument 2 Number		Parameter not used with current Control Source selected
SP Table 3 - Condition 2, Argument 1 Source	0	Control Not Used
SP Table 3 - Condition 2, Argument 1 Number		Parameter not used with current Control Source selected
SP Table 3 - Condition 2, Operator		Parameter not used with current Control Source selected
SP Table 3 - Condition 2, Argument 2 Source	0	Control Not Used
SP Table 3 - Condition 2, Argument 2 Number		Parameter not used with current Control Source selected
SP Table 3 - Condition 3, Argument 1 Source	0	Control Not Used
SP Table 3 - Condition 3, Argument 1 Number		Parameter not used with current Control Source selected
SP Table 3 - Condition 3, Operator		Parameter not used with current Control Source selected
SP Table 3 - Condition 3, Argument 2 Source	0	Control Not Used
SP Table 3 - Condition 3, Argument 2 Number		Parameter not used with current Control Source selected

**Screen Capture of Default Programmable Logic 1 Setpoints**

*Note: In the screen capture shown above, the “Programmable Logic Block Enabled” has been changed from its default value in order to enable the function block.*

*Note: The default values for the Argument1, Argument 2 and Operator are all the same across all the Programmable Logic function blocks, and must therefore be changed by the user as appropriate before this can be used.*

#### 4.9. Math Function Block Setpoints

The Math Function Block is defined in Section 1.13. Please refer to that section for detailed information on how these setpoints are used.

Setpoint Name	Value	Comment
SP Math Function Enabled	1	True
SP Function 1 Input A Source	2	Universal Input Measured
SP Function 1 Input A Number	1	Universal Input Measured #1
SP Function 1 Input A Minimum	0.00	
SP Function 1 Input A Maximum	3.75	
SP Function 1 Input A Scaler	1.00	
SP Function 1 Input B Source	1	Received CAN Message
SP Function 1 Input B Number	2	Received CAN Message #2
SP Function 1 Input B Minimum	0.00	
SP Function 1 Input B Maximum	4.50	
SP Function 1 Input B Scaler	1.00	
SP Math Function 1 Operation	9	+, Result = InA plus InB
SP Function 2 Input B Source	0	Control Not Used
SP Function 2 Input B Number		Parameter not used with current Control Source selected
SP Function 2 Input B Minimum		Parameter not used with current Control Source selected
SP Function 2 Input B Maximum		Parameter not used with current Control Source selected
SP Function 2 Input B Scaler		Parameter not used with current Control Source selected
SP Math Function 2 Operation (Input A = Result of Function 1)		Parameter not used with current Control Source selected
SP Function 3 Input B Source	0	Control Not Used
SP Function 3 Input B Number		Parameter not used with current Control Source selected
SP Function 3 Input B Minimum		Parameter not used with current Control Source selected
SP Function 3 Input B Maximum		Parameter not used with current Control Source selected
SP Function 3 Input B Scaler		Parameter not used with current Control Source selected
SP Math Function 3 Operation (Input A = Result of Function 2)		Parameter not used with current Control Source selected
SP Math Output Minimum Range	0.00	
SP Math Output Maximum Range	10.00	

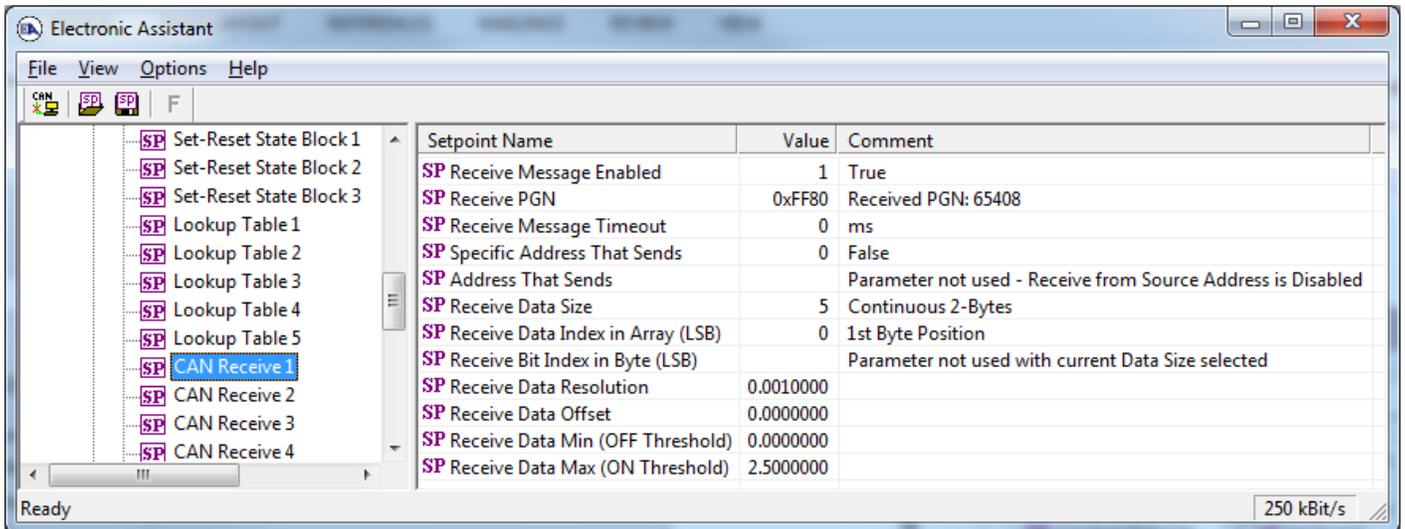
**Screen Capture of an Example for Math Function Block**

*Note: In the screen capture shown above, the setpoints have been changed from their default values to illustrate an example of how the Math Function Block can be used.*

Name	Range	Default	Notes
Math Function Enabled	Drop List	0 – FALSE	TRUE or FALSE
Function 1 Input A Source	Drop List	0 – Control Not Used	Refer to Section 1.4
Function 1 Input A Number	Depends on Source	1	Refer to Section 1.4
Function 1 Input A Minimum	-10 <sup>6</sup> to 10 <sup>6</sup>	0	Converts input to percentage before being used in calculation
Function 1 Input A Maximum	-10 <sup>6</sup> to 10 <sup>6</sup>	100	Converts input to percentage before being used in calculation
Function 1 Input A Scaler	-1.00 to 1.00	1.00	Refer to Section 1.13
Function 1 Input B Source	Drop List	0 – Control Not Used	Refer to Section 1.4
Function 1 Input B Number	Depends on Source	1	Refer to Section 1.4
Function 1 Input B Minimum	-10 <sup>6</sup> to 10 <sup>6</sup>	0	Converts input to percentage before being used in calculation
Function 1 Input B Maximum	-10 <sup>6</sup> to 10 <sup>6</sup>	100	Converts input to percentage before being used in calculation
Function 1 Input B Scaler	-1.00 to 1.00	1.00	Refer to Section 1.13
Math Function 1 Operation	Drop List	9, +, Result = InA+InB	Refer to Section 1.13
Function 2 Input B Source	Drop List	0 – Control Not Used	Refer to Section 1.4
Function 2 Input B Number	Depends on Source	1	Refer to Section 1.4
Function 2 Input B Minimum	-10 <sup>6</sup> to 10 <sup>6</sup>	0	Converts input to percentage before being used in calculation
Function 2 Input B Maximum	-10 <sup>6</sup> to 10 <sup>6</sup>	100	Converts input to percentage before being used in calculation
Function 2 Input B Scaler	-1.00 to 1.00	1.00	Refer to Section 1.13
Math Function 2 Operation (Input A = Result of Function 1)	Drop List	9, +, Result = InA+InB	Refer to Section 1.13
Function 3 Input B Source	Drop List	0 – Control Not Used	Refer to Section 1.4
Function 3 Input B Number	Depends on Source	1	Refer to Section 1.4
Function 3 Input B Minimum	-10 <sup>6</sup> to 10 <sup>6</sup>	0	Converts input to percentage before being used in calculation
Function 3 Input B Maximum	-10 <sup>6</sup> to 10 <sup>6</sup>	100	Converts input to percentage before being used in calculation
Function 3 Input B Scaler	-1.00 to 1.00	1.00	Refer to Section 1.13
Math Function 3 Operation (Input A = Result of Function 2)	Drop List	9, +, Result = InA+InB	Refer to Section 1.13
Math Output Minimum Range	-10 <sup>6</sup> to 10 <sup>6</sup>	0	
Math Output Maximum Range	-10 <sup>6</sup> to 10 <sup>6</sup>	100	

#### 4.10. CAN Receive Setpoints

The CAN Receive function block is defined in Section 1.16. Please refer there for detailed information about how all these setpoints are used.

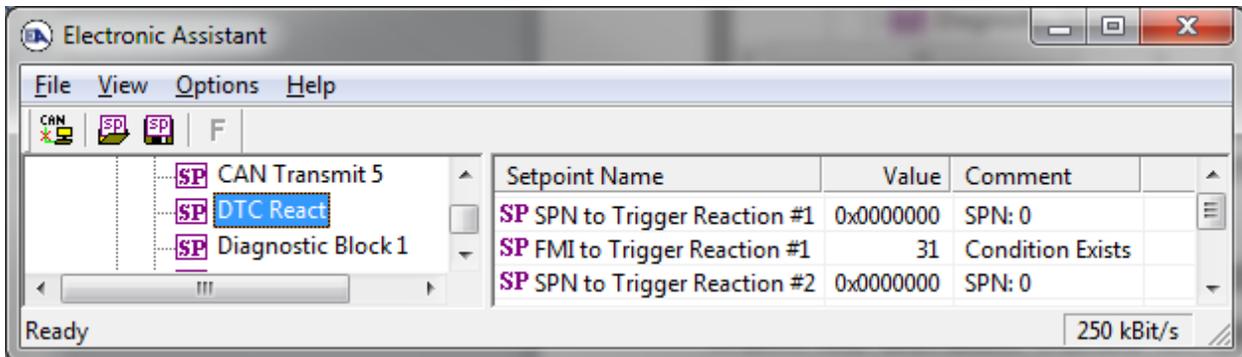


**Screen Capture of Default CAN Receive 1 Setpoints**

*Note: In the screen capture shown above, the “Receive Message Enabled” has been changed from its default value in order to enable the function block.*

#### 4.11. DTC React

The DTC React Function Block is defined in Section 1.1512. Please refer there for detailed information about how these setpoints are used.

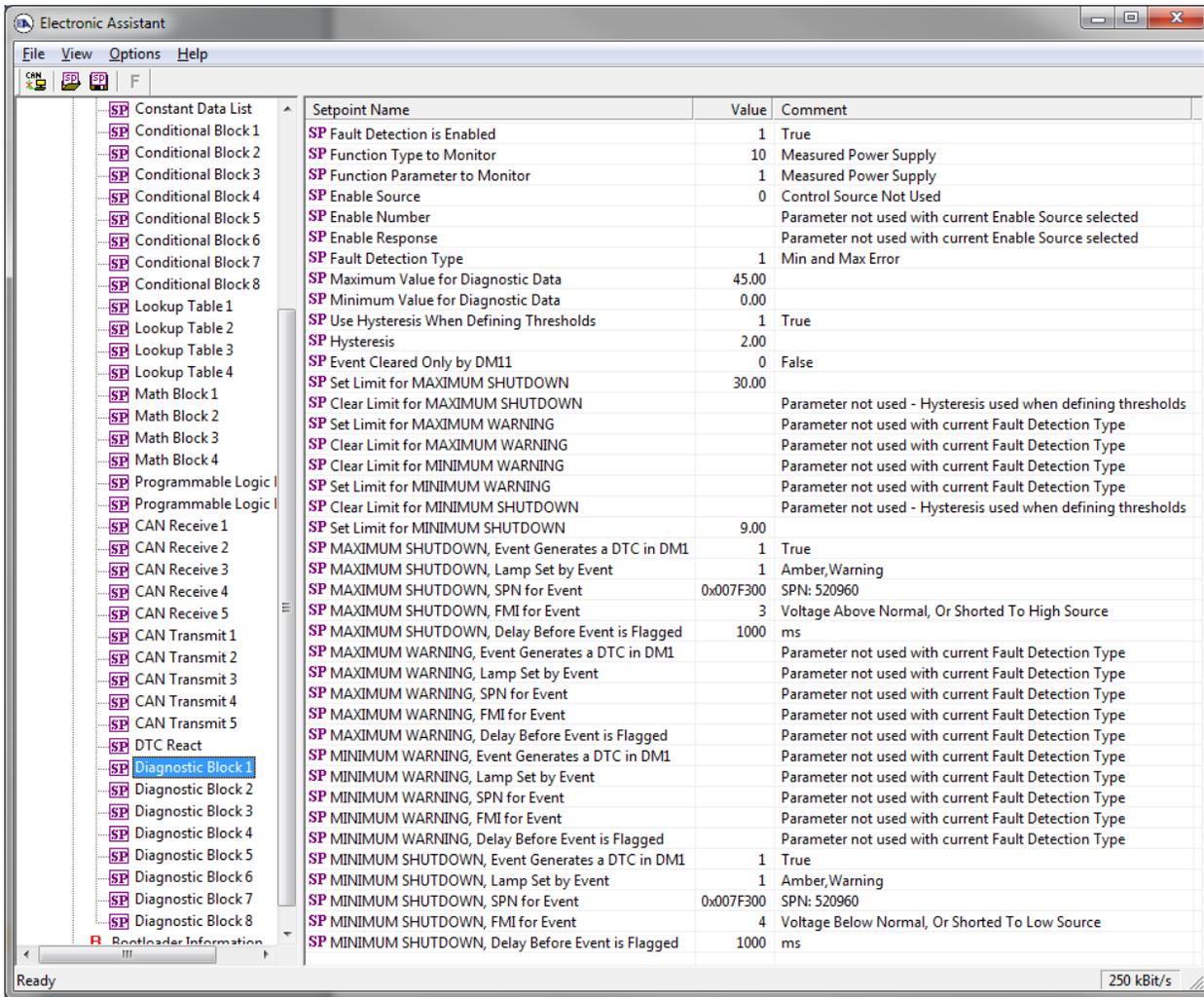


**Screen Capture of DTC Setpoints**

Name	Range	Default	Notes
SPN to Trigger Reaction #1	0 to 524287	0	0 is an illegal value, and disables the DTC
FMI to Trigger Reaction #X	Drop List	31, Condition Exists	Supports all FMIs in the J1939 standard

#### 4.12. Diagnostics Blocks

There are 8 Diagnostics blocks that can be configured to monitor various parameters of the Controller. The Diagnostic Function Block is defined in section 1.76. Please refer there for detailed information how these setpoints are used.



### Screen Capture of Default Diagnostic Block 1 Setpoints

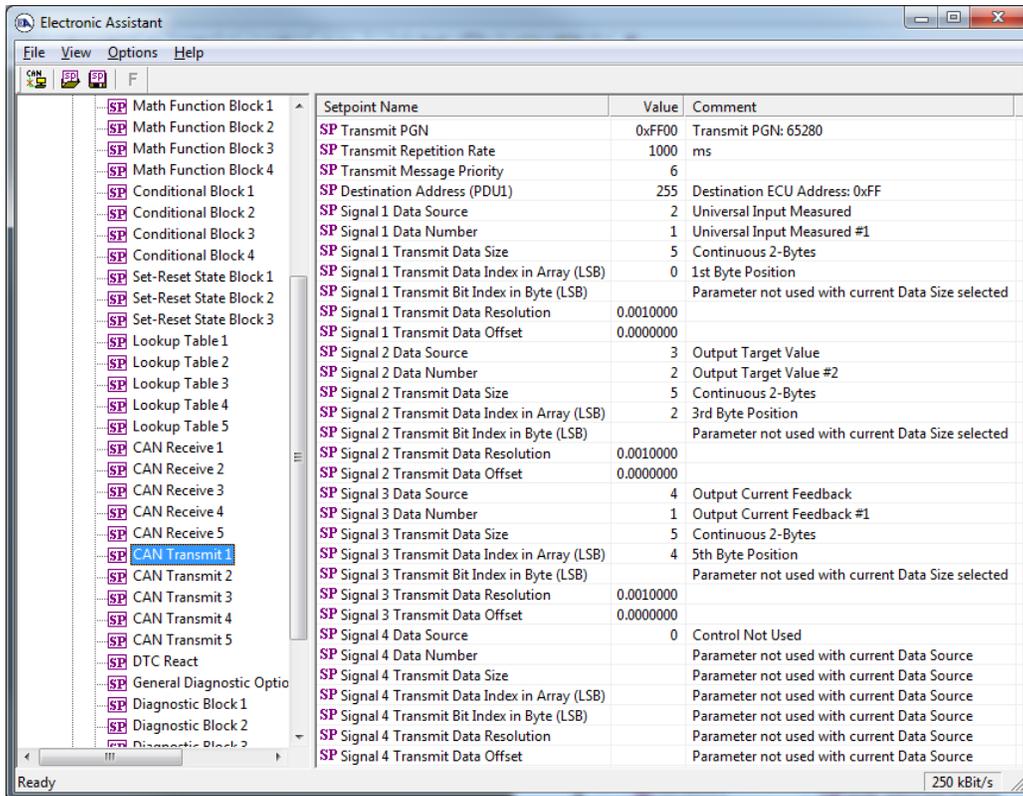
Name	Range	Default	Notes
Fault Detection is Enabled	Drop List	False	
Function Type to Monitor	Drop List	0 – Control not used	
Function parameter to Monitor	Drop List	0 – No selection	
Fault Detection Type	Drop List	1 – Min and Max Error	See section 1.7
Maximum Value for Diagnostic Data	Minimum Value for Diagnostic Data ... 4.28e <sup>9</sup>	5.0	
Minimum Value for Diagnostic Data	0.0 ... Maximum Value for Diagnostic Data	0.0	
Use Hysteresis When Defining Thresholds	Drop List	False	
Hysteresis	0.0 ... Maximum Value for Diagnostic Data	0.0	
Event Cleared only by DM11	Drop List	False	
Set Limit for MAXIMUM SHUTDOWN	Minimum Value for Diagnostic Data ...	4.8	

	Maximum Value for Diagnostics Data		
Clear Limit for MAXIMUM SHUTDOWN	Minimum Value for Diagnostic Data ... Maximum Value for Diagnostics Data	4.6	
Set Limit for MAXIMUM WARNING	Minimum Value for Diagnostic Data ... Maximum Value for Diagnostics Data	0.0	
Clear Limit for MAXIMUM WARNING	Minimum Value for Diagnostic Data ... Maximum Value for Diagnostics Data	0.0	
Clear Limit for MINIMUM WARNING	Minimum Value for Diagnostic Data ... Maximum Value for Diagnostics Data	0.0	
Set Limit for MINIMUM WARNING	Minimum Value for Diagnostic Data ... Maximum Value for Diagnostics Data	0.0	
Clear Limit for MINIMUM SHUTDOWN	Minimum Value for Diagnostic Data ... Maximum Value for Diagnostics Data	0.4	
Set Limit for MINIMUM SHUTDOWN	Minimum Value for Diagnostic Data ... Maximum Value for Diagnostics Data	0.2	
MAXIMUM SHUTDOWN, Event Generates a DTC in DM1	Drop List	True	
MAXIMUM SHUTDOWN, Lamp Set by Event	Drop List	0 – Protect	See Table
MAXIMUM SHUTDOWN, SPN for Event	0...524287	520448 (\$7F100)	It is the user's responsibility to select an SPN that will not violate the J1939 standard.
MAXIMUM SHUTDOWN, FMI for Event	Drop List	3, Voltage Above Normal	See Table
MAXIMUM SHUTDOWN, Delay Before Event is Flagged	0...60000 ms	1000	
MAXIMUM WARNING, Event Generates a DTC in DM1	Drop List	True	
MAXIMUM WARNING, Lamp Set by Event	Drop List	0 – Protect	See Table
MAXIMUM WARNING, SPN for Event	0...524287	520704 (\$7F200)	It is the user's responsibility to select an SPN that will not violate the J1939 standard.
MAXIMUM WARNING, FMI for Event	Drop List	3, Voltage Above Normal	See Table
MAXIMUM WARNING, Delay Before Event is Flagged	0...60000 ms	1000	
MINIMUM WARNING, Event Generates a DTC in DM1	Drop List	True	
MINIMUM WARNING, Lamp Set by	Drop List	0 – Protect	See Table

Event			
MAXIMUM WARNING, SPN for Event	0...524287	520960 (\$7F300)	It is the user's responsibility to select an SPN that will not violate the J1939 standard.
MINIMUM WARNING, FMI for Event	Drop List	4, Voltage Below Normal	See Table
MINIMUM WARNING, Delay Before Event is Flagged	0...60000 ms	1000	
MINIMUM SHUTDOWN, Event Generates a DTC in DM1	Drop List	True	
MINIMUM SHUTDOWN, Lamp Set by Event	Drop List	Amber Warning	See Table
MINIMUM SHUTDOWN, SPN for Event	0...524287	521216 (\$7F400)	It is the user's responsibility to select an SPN that will not violate the J1939 standard.
MINIMUM SHUTDOWN, FMI for Event	Drop List	4, Voltage Below Normal	See Table
MINIMUM SHUTDOWN, Delay Before Event is Flagged	0...60000 ms	1000	

### 4.13. CAN Transmit Setpoints

The CAN Transmit function block is defined in Section 1.11. Please refer there for detailed information about how all these setpoints are used.



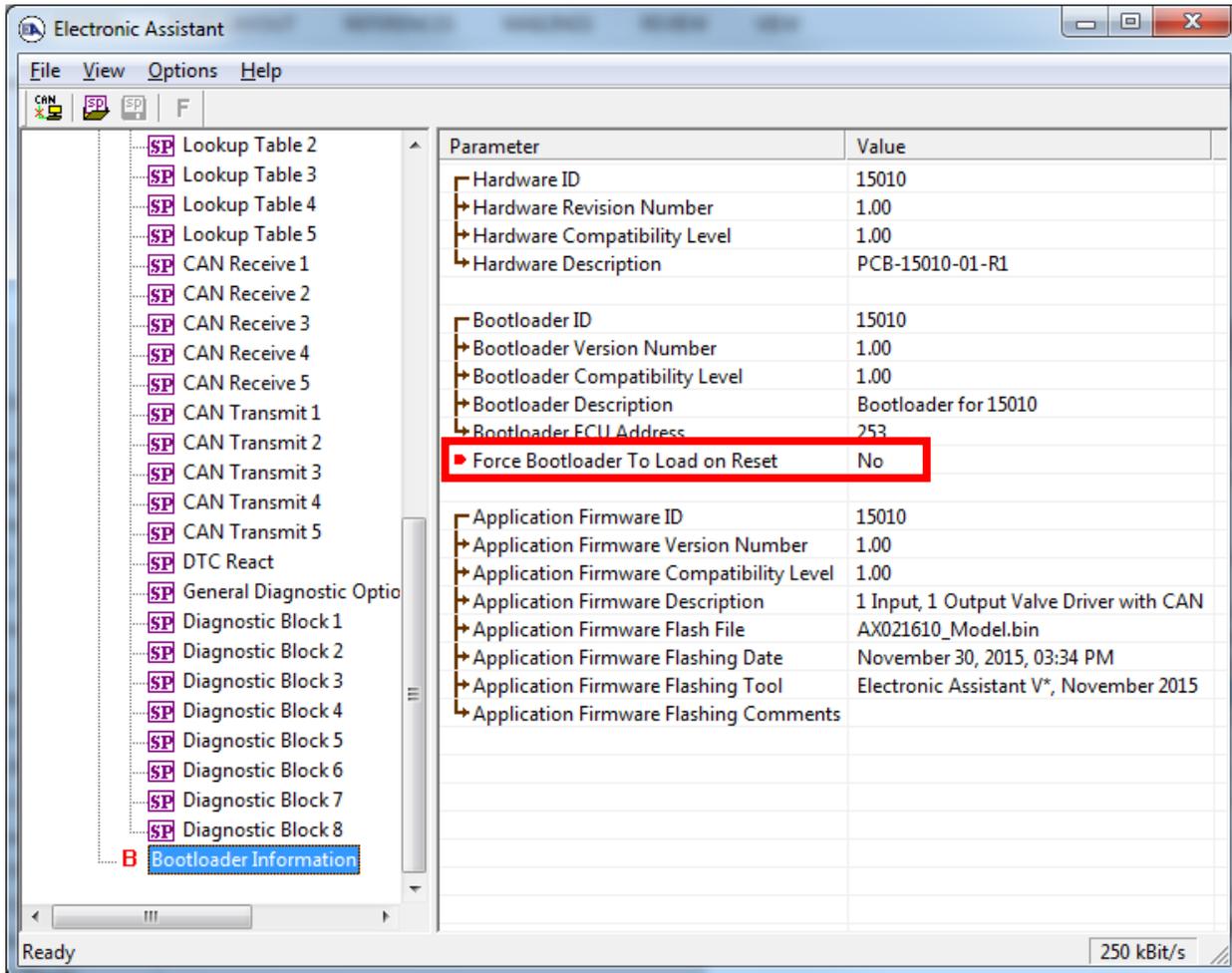
Screen Capture of Default CAN Transmit 1 Setpoints

Name	Range	Default	Notes
Transmit PGN	0 to 65535	65280 (\$FF00)	
Transmit Repetition Rate	0 to 60,000 ms	0	0ms disables transmit
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 Source	Drop List	Different for each	Refer to Section 1.4
Transmit Data Number	Per Source	Different for each	Refer to Section 1.4
Transmit Data Size	Drop List	Continuous 2-Bytes	0 = Not Used (disabled) 1 = 1-Bit 2 = 2-Bits 3 = 4-Bits 4 = 1-Byte 5 = 2-Bytes 6 = 4-Bytes
Transmit Data Index in Array (LSB)	1 to 9-DataSize	Different for each	
Transmit Bit Index in Byte (LSB)	1 to 9-BitSize	Different for each	Only used with Bit Data Types
Transmit Data Resolution	-10 <sup>6</sup> to 10 <sup>6</sup>	Different for each	
Transmit Data Offset	-10 <sup>4</sup> to 10 <sup>4</sup>	Different for each	

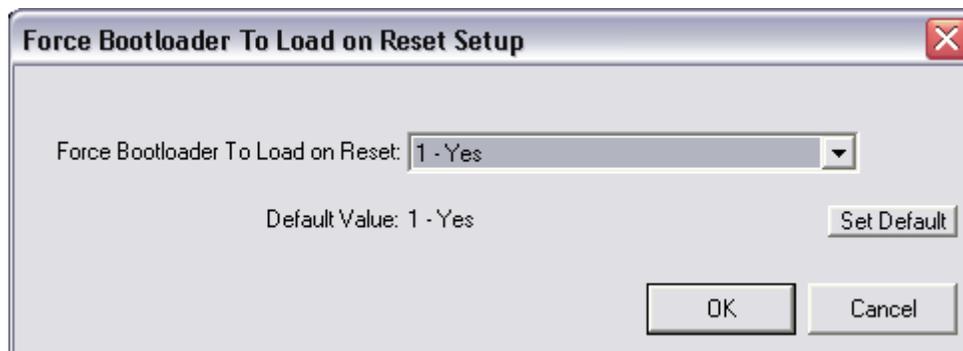
## 5. REFLASHING OVER CAN WITH THE AXIOMATIC EA BOOTLOADER

The AX021610 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.

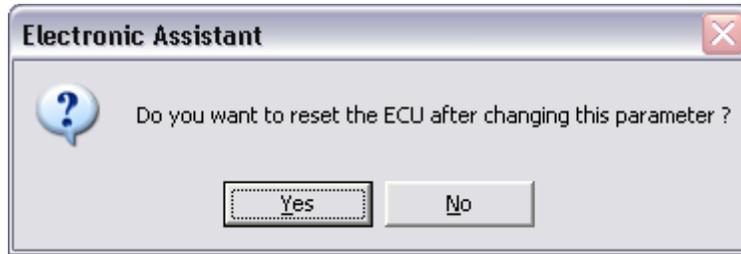
1. When the Axiomatic EA first connects to the ECU, the **Bootloader Information** section will display the following information.



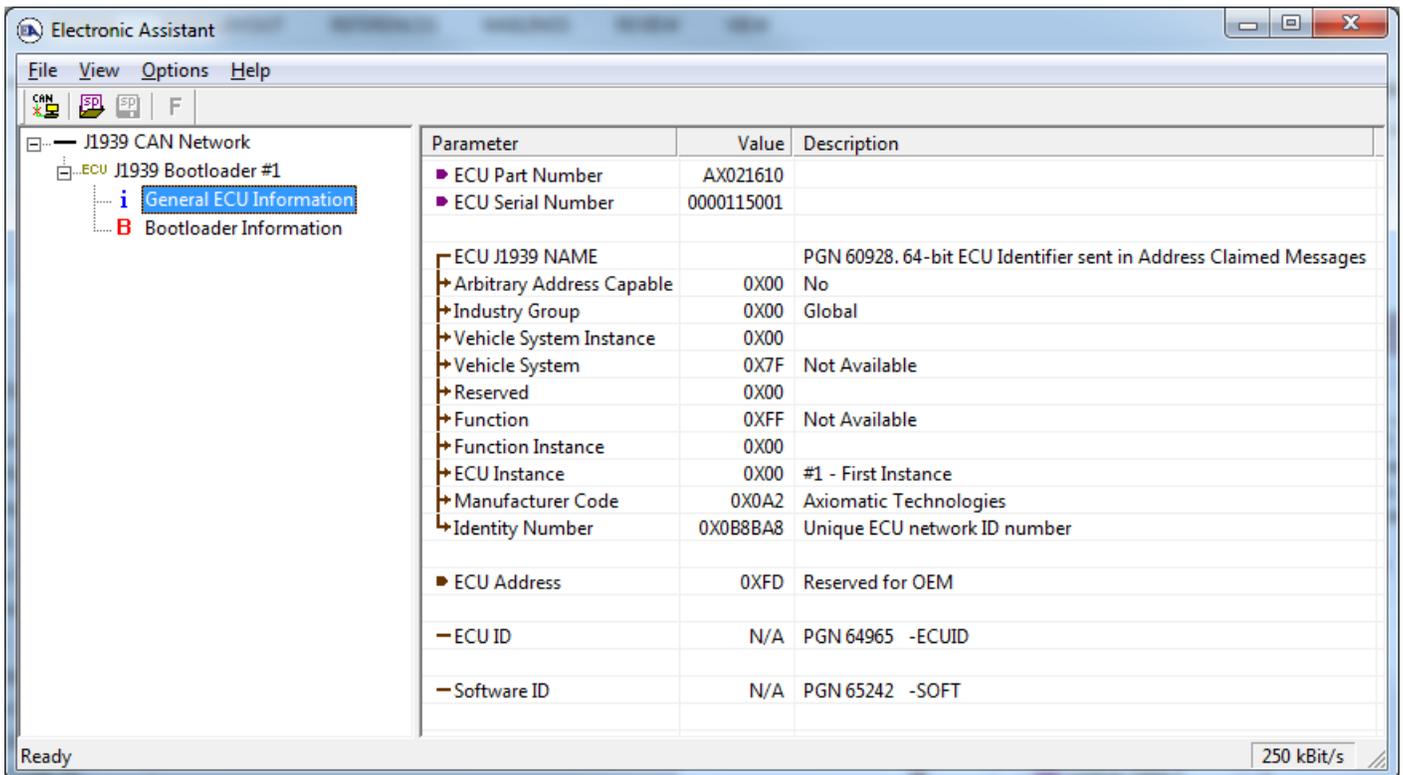
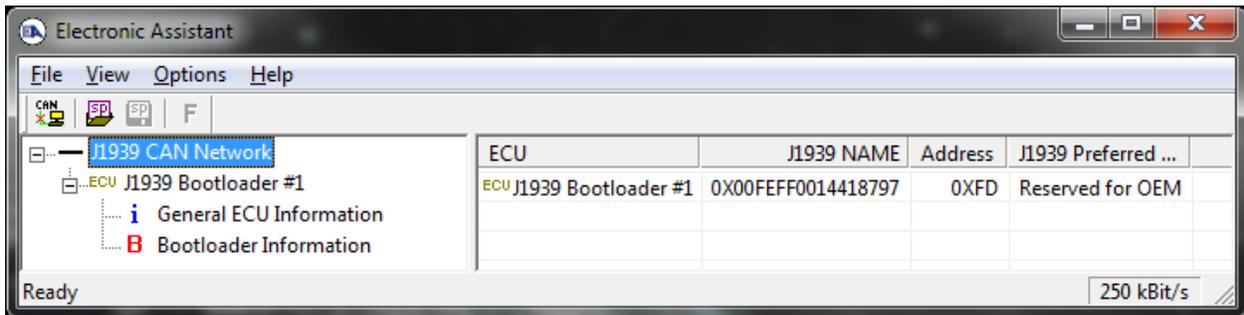
2. To use the bootloader to upgrade the firmware running on the ECU, change the variable **“Force Bootloader To Load on Reset”** to Yes.



3. When the prompt box asks if you want to reset the ECU, select Yes.

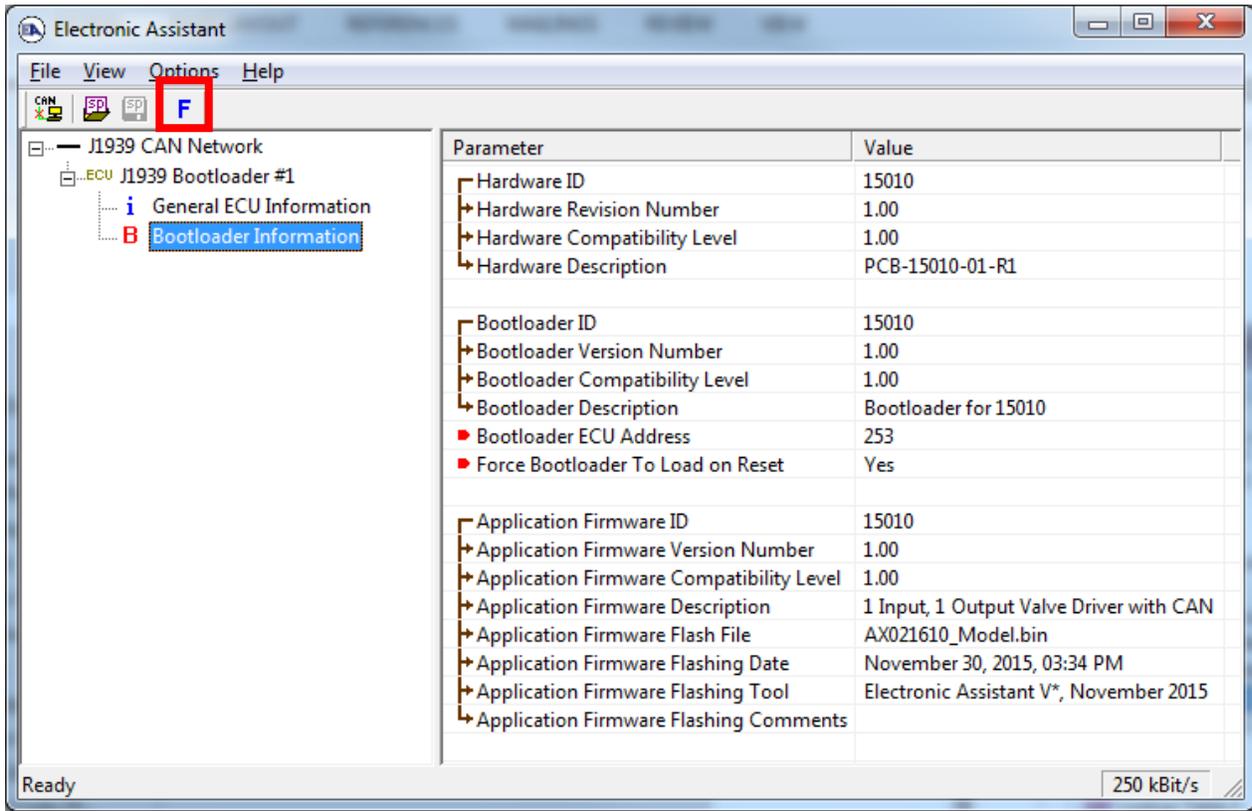


4. Upon reset, the ECU will no longer show up on the J1939 network as an AX021610 but rather as **J1939 Bootloader #1**.



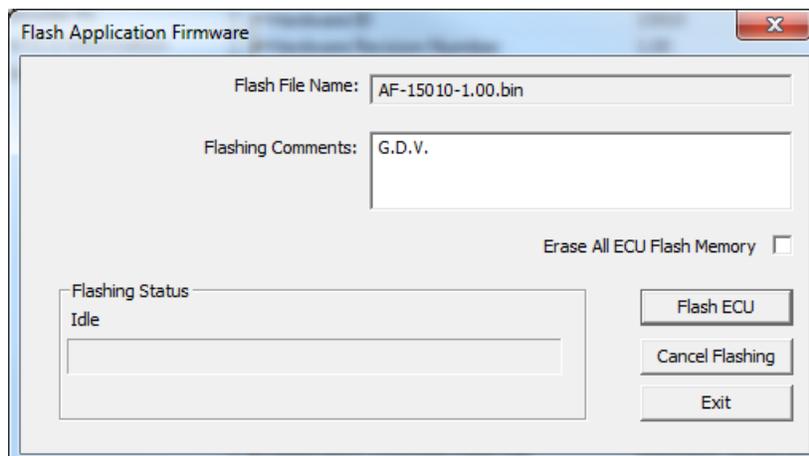
*Note that the bootloader is NOT Arbitrary Address Capable. This means that if you want to have multiple bootloaders running simultaneously (not recommended) you would have to manually change the address for each one before activating the next, or there will be address conflicts, and only one ECU would 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.*

- When the **Bootloader Information** section is selected, the same information is shown as when it was running the AX021610 firmware, but in this case the **Flashing** feature has been enabled.



- Select the **Flashing** button and navigate to where you had saved the **AF-15010-x.yy.bin** file sent from Axiomatic. (Note: only binary (.bin) files can be flashed using the Axiomatic EA tool)
- 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.

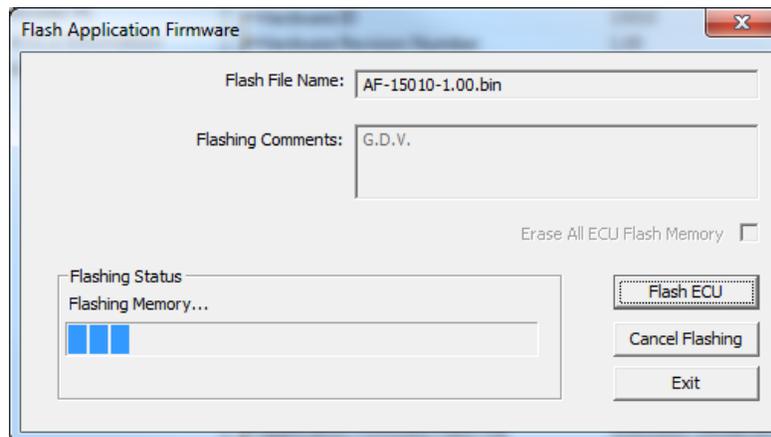
Note: You do not have to date-stamp or timestamp the file, as this is all done automatically by the Axiomatic EA tool when you upload the new firmware.



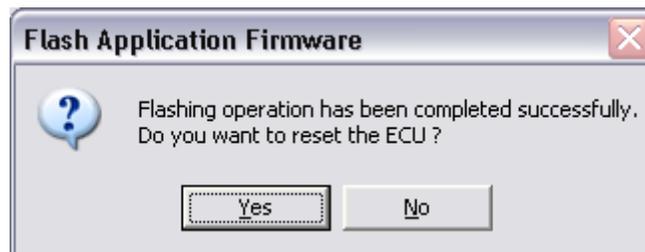


**WARNING:** Do not check the “Erase All ECU Flash Memory” box unless instructed to do so by your Axiomatic contact. Selecting this will erase ALL data stored in non-volatile flash. It will also erase any configuration of the setpoints that might have been done to the ECU and reset all setpoints to their 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 AX021610 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 AX021610 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.

## 6. 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/>.

### 6.1. Power Supply

Power Supply Input - Nominal	12 or 24Vdc nominal operating voltage 8..36 Vdc power supply range for voltage transients Overvoltage protection up to 71V is provided. Overvoltage (undervoltage) shutdown of the output load is provided.
Surge Protection	Provided
Reverse Polarity Protection	Provided

### 6.2. Input

Analog Input Functions	Voltage Input, Current Input or Resistive Input
Voltage Input	0-1V (Impedance 1 MOhm) 0-2.5V (Impedance 1 MOhm) 0-5V (Impedance 200 KOhm) 0-10V (Impedance 133 KOhm for 0-5V, 133 to 20 KOhm for 5-10V)
Current Input	0-20 mA (Impedance 124 Ohm) 4-20 mA (Impedance 124 Ohm)
Resistive Input	25Ω to 250 kΩ
Digital Input Functions	Discrete Input, PWM Input, Frequency, Counter Input
Digital Input Level	Up to Vps
PWM Input	0 to 100% 10 Hz to 1kHz 100 Hz to 10 kHz
Frequency Input	10 Hz to 1kHz 100 Hz to 10 kHz
Digital Input	Active High (to +Vps), Active Low Amplitude: 0 to +Vps
Counter Input	0Hz to 10kHz
Input Impedance	1 MOhm High impedance, 10KOhm pull down, 10KOhm pull up to +14V
Input Accuracy	< 1%
Input Resolution	12-bit

### 6.3. Output

Output	Up to 3A Half-bridge, High Side Sourcing, Current Sensing, Grounded Load High Frequency (25 kHz) The user can select the following options for output using the Axiomatic EA. <ul style="list-style-type: none"> <li>• Output Disable</li> <li>• Output Current (PID loop, with current sensing) (0-3A)</li> <li>• Hotshot Digital</li> <li>• Proportional Output Voltage (up to Vps)</li> <li>• Output PWM Duty Cycle (0-100% D.C.)</li> <li>• Digital On/Off (Gnd-Vps)</li> </ul>
Output Accuracy	Output Current mode $\leq 1\%$ Output Voltage mode $\leq 5\%$ Output PWM Duty Cycle mode $\leq 0.1\%$
Output Resolution	Output Current mode 1 mA Output Voltage mode 0.1V Output PWM mode 0.1%
Protection	Over-current and short circuit protection

### 6.4. Communication

CAN	1 CAN 2.0B port, protocol SAE J1939
Network Termination	According to the CAN standard, it is necessary to terminate the network with external termination resistors. The resistors are 120 Ohm, 0.25W minimum, metal film or similar type. They should be placed between CAN_H and CAN_L terminals at both ends of the network.

## 6.5. General Specifications

Microprocessor	STM32F205RET6 32-bit, 512 Kbit program flash																		
Quiescent Current	Contact Axiomatic.																		
LED Indicator	User configurable to react to different events or faults																		
Response Time	Contact Axiomatic.																		
Control Logic	User programmable functionality using the Axiomatic Electronic Assistant																		
Communications	1 CAN port (SAE J1939), CANopen® is available on request.																		
User Interface	The Axiomatic Electronic Assistant for <i>Windows</i> operating systems 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 <i>Windows</i> -based PC. An Axiomatic USB-CAN Converter is part of the Axiomatic Electronic Assistant Configuration KIT, ordering P/Ns: <b>AX070502</b> or <b>AX070506K</b> .																		
Network Termination	It is necessary to terminate the network with external termination resistors. The resistors are 120 Ohm, 0.25W minimum, metal film or similar type. They should be placed between CAN_H and CAN_L terminals at both ends of the network.																		
Operating Conditions	-40 to 85 °C (-40 to 185 °F)																		
Enclosure	Molded Enclosure, integral connectors Nylon 6/6, 30% glass Ultrasonically welded 3.47 x 2.75 x 1.31 inches (88.2 x 70.0 x 33.3 mm) L x W x H including integral connectors Refer to the dimensional drawing.																		
Protection	IP67																		
Vibration	MIL-STD-202G, Method 204D test condition C (Sine) and Method 214A, test condition B (Random) 10 g peak (Sine) 7.68 Grms peak (Random)																		
Shock	MIL-STD-202G, Method 213B, test condition A 50g (half sine pulse, 9ms long, 8 per axis)																		
Approvals	CE marking																		
Weight	0.156 lb. (0.071 kg)																		
Electrical Connections	Integral 8-pin receptacle (equivalent TE Deutsch P/N: DT04-08PA) 18 AWG wire is recommended for use with contacts 0462-201-16141.  A mating plug kit is available. Ordering P/N: <b>AX070112</b> is comprised of 1 DT06-08SA, 1 W8S, 8 0462-201-16141, and 3 114017.  <table border="1"> <thead> <tr> <th>PIN #</th> <th>FUNCTION</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>CAN_L</td> </tr> <tr> <td>2</td> <td>CAN_H</td> </tr> <tr> <td>3</td> <td>Output GND</td> </tr> <tr> <td>4</td> <td>Universal Input</td> </tr> <tr> <td>5</td> <td>Input Analog GND</td> </tr> <tr> <td>6</td> <td>Output +</td> </tr> <tr> <td>7</td> <td>Batt -</td> </tr> <tr> <td>8</td> <td>Batt +</td> </tr> </tbody> </table>	PIN #	FUNCTION	1	CAN_L	2	CAN_H	3	Output GND	4	Universal Input	5	Input Analog GND	6	Output +	7	Batt -	8	Batt +
PIN #	FUNCTION																		
1	CAN_L																		
2	CAN_H																		
3	Output GND																		
4	Universal Input																		
5	Input Analog GND																		
6	Output +																		
7	Batt -																		
8	Batt +																		
Mounting	Mounting holes are sized for #8 or M4 bolts. The bolt length will be determined by the end-user's mounting plate thickness. The mounting flange of the controller is 0.425 inches (10.8 mm) thick.  If the module is mounted without an enclosure, it should be mounted vertically with connectors facing left or right to reduce likelihood of moisture entry.  The CAN wiring is considered intrinsically safe. The power wires are not considered intrinsically safe and so in hazardous locations, they need to be located in conduit or conduit trays at all times. The module must be mounted in an enclosure in hazardous locations for this purpose.  No wire or cable harness should exceed 30 meters in length. The power input wiring should be limited to 10 meters.  All field wiring should be suitable for the operating temperature range.  Install the unit with appropriate space available for servicing and for adequate wire harness access (6 inches or 15 cm) and strain relief (12 inches or 30 cm).																		

## 7. VERSION HISTORY

---

<b>Version</b>	<b>Date</b>	<b>Author</b>	<b>Modifications</b>
1	November 30 <sup>th</sup> , 2015	Gustavo Del Valle	Initial Draft
-	December 18 <sup>th</sup> , 2015	Amanda Wilkins	Marketing Review
-	December 30 <sup>th</sup> , 2015	Amanda Wilkins	Added vibration compliance
1B	March 31 <sup>st</sup> , 2016	Amanda Wilkins	Added CE marking
2	July 11 <sup>th</sup> , 2016	Gustavo Del Valle	Reflects additional setpoints added to the output function block. Firmware supports configuration of the output regulator PID parameters for tuning. Sections updated were Section 1.6 and Section 4.5  Reflects additional setpoints for Digital ON Time and Digital OFF Time control sources and numbers. Updated description of these setpoints  Fixed type in the Universal Input section
-	October 24, 2016	Amanda Wilkins	Updated dimensional drawing with final labels
2A	August 8, 2023	Kiril Mojsov	Performed Legacy Updates

**Note:**

CANopen® is a registered community trademark of CAN in Automation e.V.

## 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](http://axiomatic.com). Any inquiries should be sent to [sales@axiomatic.com](mailto: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](http://www.P65Warnings.ca.gov).

## SERVICE

All products to be returned to Axiomatic require a Return Materials Authorization Number (RMA#) from [sales@axiomatic.com](mailto: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](http://www.axiomatic.com)  
[sales@axiomatic.com](mailto:sales@axiomatic.com)

**Axiomatic Technologies Oy**  
Höytämöntie 6  
33880 Lempäälä  
FINLAND  
TEL: +358 103 375 750  
[www.axiomatic.com](http://www.axiomatic.com)  
[salesfinland@axiomatic.com](mailto:salesfinland@axiomatic.com)