USER MANUAL UMAX031500A
Version V1.0.1

Magnetic Pick-Up Input To Can Controller

SAEJ1939

USER MANUAL

P/N: AX031500A

Axiomatic Technologies Oy
Höytämöntie 6
33880 LEMPÄÄLÄ, Finland
Tel. +358 103 375 750
salesfinland@axiomatic.com
www.axiomatic.fi

Axiomatic Technologies Corporation
1445 Courtneypark Dr. E.
Mississauga, ON Canada L5T 2E3
Tel. 1 905 602 9270
sales@axiomatic.com
www.axiomatic.com

ACCRONYMS

ACK	Positive Acknowledgement (from SAE J1939 standard)
MAGIN	Magnetic Pick-up Input
EA	Axiomatic Electronic Assistant (A Service Tool Kit 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)

TABLE OF CONTENTS

1. OVERVIEW OF CONTROLLER

1.1. Description of Magnetic Pick-Up Input Controller

The Magnetic Pick-up Input to CAN Controller (MAGIN-CAN) is designed for versatile control of a single magnetic pick-up input and a wide variety of control logic and algorithms.

The magnetic pick-up input and all logical function blocks on the unit are inherently independent from one another but can be configured to interact with each other in a number of ways.

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

1.2. Magnetic Pick-up Input Function Block

The controller consists of a single magnetic pick-up input. Although the physical input is fixed, it can be used in a number of ways through software as listed in Section 1.2.1 via various **Input Sensor Types.**

1.2.1. Magnetic Pick-up Input Sensor Types

The **Input Sensor Type** parameter provides a dropdown list with the input types described in Table 1. Changing the **Input Sensor Type** affects other setpoints within the same setpoint group such as Minimum/Maximum Error/Range by refreshing them based on the new input type and thus should be changed first.

0	*Disabled*
40	*Frequency 0.5 to 50kHz*
70	*Counter (Pulse Count Reset)*
71	*Counter (Time Window)*
72	*Counter (Pulse Window)*

Table 1 – Universal Input Sensor Type Options

Frequency/RPM and Counter **Input Sensor Types** are connected to the microcontroller timers.

The setpoint **Pulses per Revolution** is only taken into consideration when the **Input Sensor Type** selected is *Frequency 0.5 to 50kHz* mode. 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].

There are three different types of a *Counter* input mode: *Counter (Pulse Count Reset), Counter (Time Window), Counter (Pulse Window).*

Counter (Pulse Count Reset) is a Counter input type in which the output data of the input function block is the number of pulses measured. The pulses will continue to increment in count as they are measured until the **Pulse Count to Reset Counter** value is reached at which the counter will reset back to 0.
Counter (Time Window) is a Counter input type in which a **Pulse Count in Time Window** is selected (in milliseconds). The output of the input block will be the amount of pulses measured within that time frame.

Counter (Pulse Window) is a Counter input type in which a 'Pulse Window' is selected (in number of pulses). The output of the input block will be the amount of time elapsed to reach the number of pulses set in the **Elapsed Time to Each Pulse Count** setpoint.

1.2.2. Minimum and Maximum Errors and Ranges

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

The **Minimum Error** and **Maximum Error** setpoints are used with the Diagnostic function block – please refer to Section 1.9 for more details on Diagnostic function block. The values for these setpoints are constrained such that

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

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

- Frequency: Max = 10,000 [Hz or RPM]

In order to avoid causing false faults, the user can choose to add software filtering to the measure signal.

1.2.3. Input Software Filter Types

All input types with the exception of the *Counter* Input types can be filtered using **Filter Type** and **Filter Constant** setpoints. There are three (3) filter types available as listed in Table 2.

0	*No Filtering*
1	*Moving Average*
2	*Repeating Average*

Table 2 – 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:

$$\text{Value}_N = \text{Value}_{N-1} + \frac{(\text{Input} - \text{Value}_{N-1})}{\text{Filter Constant}}$$

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:

$$\text{Value} = \frac{\sum_0^N \text{Input}_N}{N}$$

1.3. Internal Function Block Control Sources

The MAGIN-CAN 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 3.

Value	Meaning
0	*Control Source Not Used*
1	*CAN Receive Message*
2	*Magnetic Pick-up Input Measured*
3	*Frequency Measured*
4	*Lookup Table Function Block*
5	*Programmable Logic Function Block*
6	*Mathematical Function Block*
7	*Constant Data List Block*
8	*Measured Power Supply*
9	*Measured Processor Temperature*

Table 3 – 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 4 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]
CAN Receive Message	[1…8]
Magnetic Pick-up Input Measured	[1…1]
Frequency Measured	[1…1]
Lookup Table Function Block	[1…6]
Programmable Logic Function Block	[1…2]
Mathematical Function Block	[1…4]
Constant Data List Block	[1…10]
Measured Power Supply	[1…1]
Measured Processor Temperature	[1…1]

Table 4 – Control Source Number Options

The control source *Frequency Measured* differs from *Magnetic Pick-up Input Measured* in that it only transmits the measured frequency (if any) in Hertz. The *Magnetic Pick-up Input Measured* data can be in terms of frequency, RPM, pulse count, pulse duration based on *Input Sensor Type* selected from Table 1. This additional information can be useful when the *Magnetic Pick-up Input* is configured as *Counter (Pulse Count Reset)* where the user can see the count increment as well as at what frequency they are being detected by the controller.

1.4. 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 1.4.1 through 1.4.5 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 1.5.

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.

1.4.1. 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_{10} is changed first, then lower indexes in descending order as to maintain the below:

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

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 1.4.3, they will not be used in the X-Axis calculation shown above. For example, if points X_4 and higher are ignored, the formula becomes $Xmin <= X_0 <= X_1 <= X_2 <= X_3 <= Xmax$ instead.

1.4.2. 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 1.4.3, they will not be used in the Y-Axis range determination. Only the Y-Axis values shown on 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.

1.4.3. 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.

1.4.4. 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 1. To get the linearization, each "**Point N – Response**", where N = 1 to 10, is setup for a *'Ramp To'* output response.

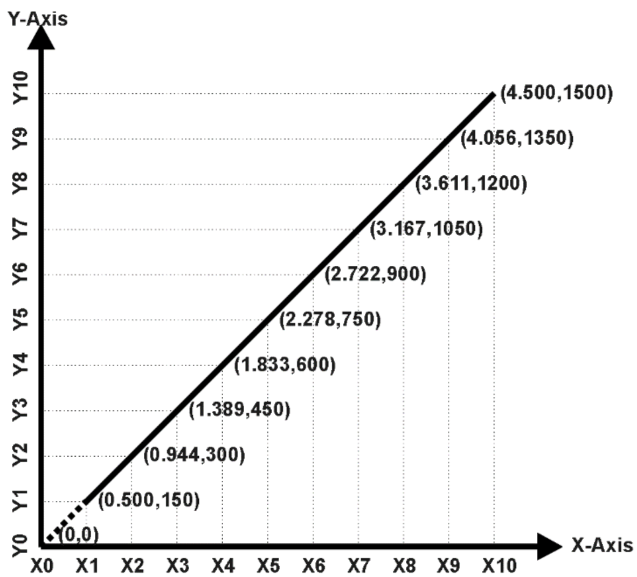


Figure 1 – Lookup Table with "Ramp To" Data Response

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 2.

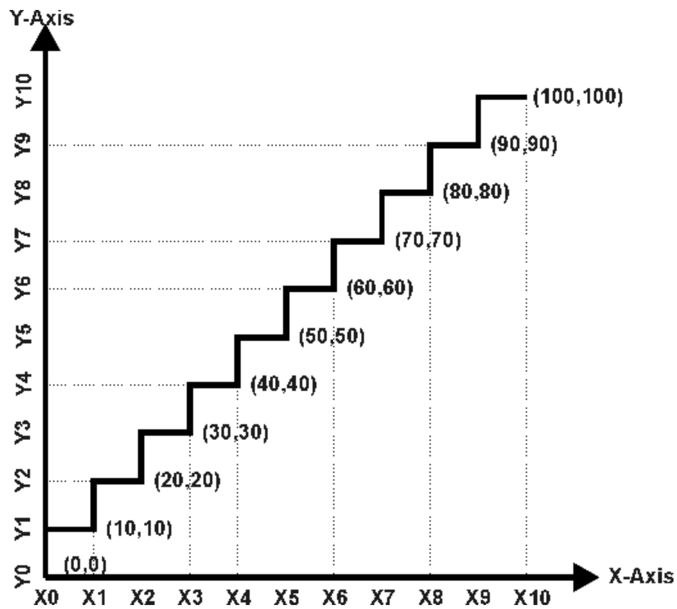


Figure 2 – Lookup Table with "Jump To" Data Response

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.

1.4.5. X-Axis, Time Response

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.

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. When the control input is ON, the output will be changed over a period of time based on the profile in the Lookup Table.

When the control input is OFF, the output is always at zero. When the input comes ON, the profile ALWAYS starts at position (X_0, Y_0) 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].

1.5. Programmable Logic Function Block

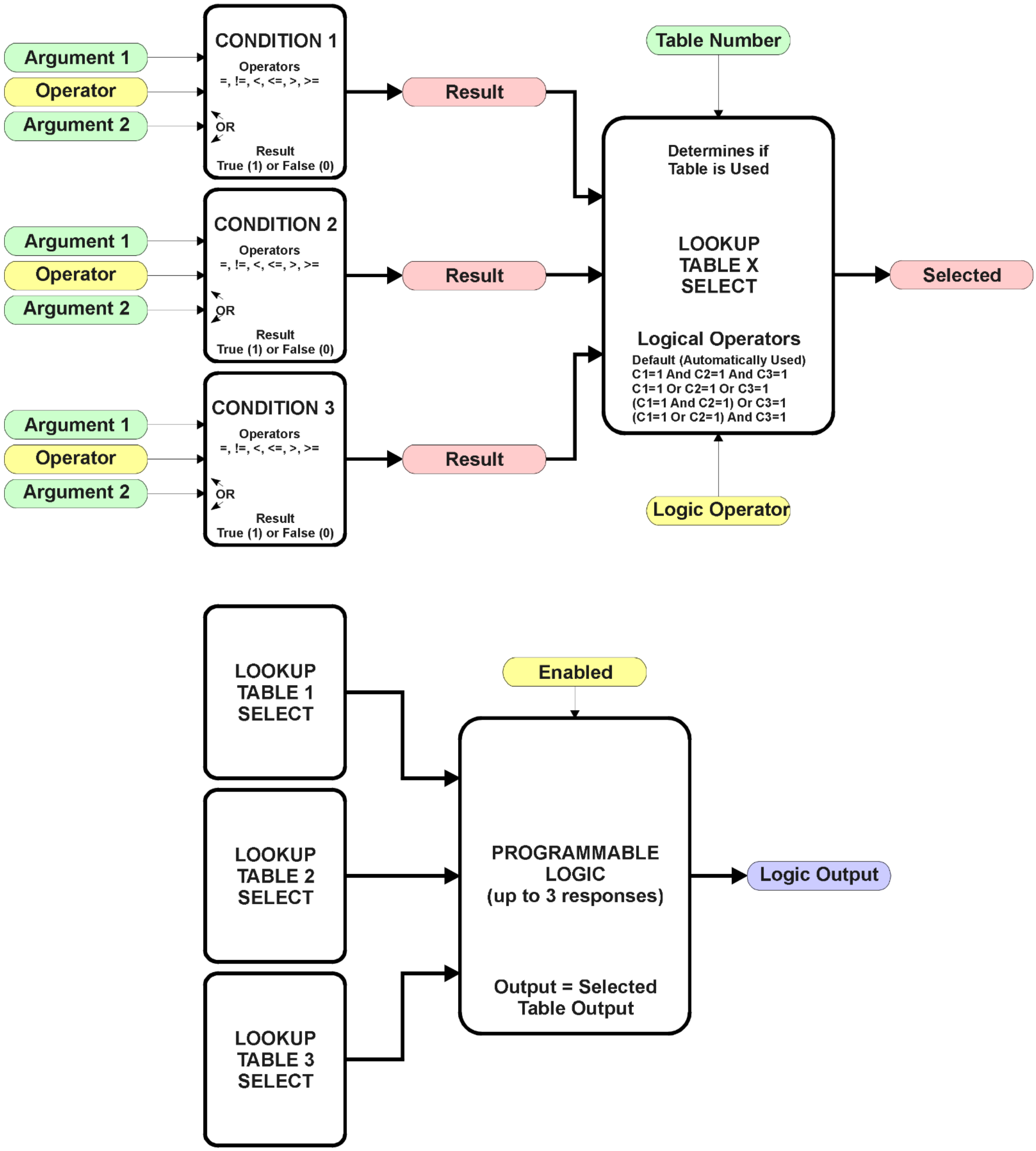


Figure 3 – Programmable Logic Function Block

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 1.5.2, 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 4. 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.**

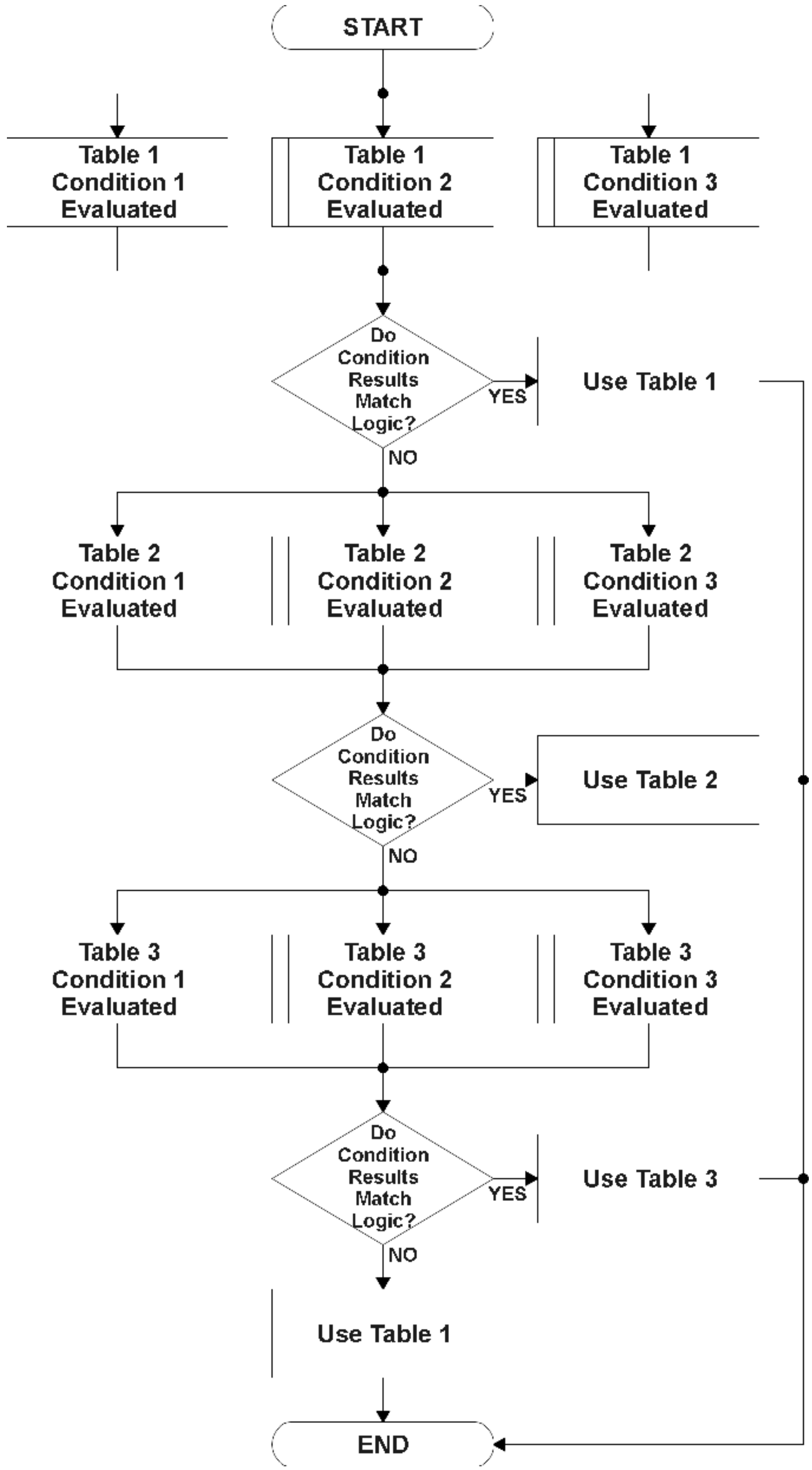


Figure 4 – Programmable Logic Flowchart

1.5.1. 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 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 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 5.

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

Table 5 – Condition Operator Options

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 4 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 6.

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 *'Control Source Not Used'*)

Table 6– Condition Evaluation Results

1.5.2. 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 1.5.1. There are several logical combinations that can be selected, as listed in Table 7.

0	*Default Table*
1	*Cnd1 And Cnd2 And Cnd3*
2	*Cnd1 Or Cnd2 Or Cnd3*
3	*(Cnd1 And Cnd2) Or Cnd3*
4	*(Cnd1 Or Cnd2) And Cnd3*

Table 7 – 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.

Logical Operator	**Select Conditions Criteria**
Default Table	Associated table is automatically selected as soon as it is evaluated.
Cnd1 And Cnd2 And Cnd3	**Should be used when two or three conditions are relevant, and all must be true to select the table.** 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. If((Cnd1==True) &&(Cnd2==True)&&(Cnd3==True)) Then Use Table
Cnd1 Or Cnd2 Or Cnd3	**Should be used when only one condition is relevant. Can also be used with two or three relevant conditions.** If any condition is evaluated as True, the table is selected. Error or N/A results are treated as False If((Cnd1==True) || (Cnd2==True) || (Cnd3==True)) Then Use Table
(Cnd1 And Cnd2) Or Cnd3	**To be used only when all three conditions are relevant.** 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 If(((Cnd1==True)&&(Cnd2==True)) || (Cnd3==True)) Then Use Table
(Cnd1 Or Cnd2) And Cnd3	**To be used only when all three conditions are relevant.** 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 If(((Cnd1==True)||(Cnd2==True)) && (Cnd3==True)) Then Use Table

Table 8 – 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.'*

1.5.3. 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 9.

Programmable Logic Block Number	Table 1 – Lookup Table Block Number	Table 2 – Lookup Table Block Number	Table 3 – Lookup Table Block Number
1	1	2	3

Table 9 – 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.

1.6. 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 $\mathrm{A\ operator\ B}$, where A and B are function inputs and operator is function selected with setpoint **Math function X Operator**. Setpoint options are presented in Table 11. 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.

$$Math\ Block\ Output = \Big(\big((A1\ op1\ B1)op2\ B2\big)op3\ B3\ \Big)\ op4\ B4$$

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

Table 10 – 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 is 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.

1.7. CAN Transmit Function Block

The CAN Transmit function block is used to send any output from another function block (i.e. input, logic signal) 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.

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

The MAGIN-CAN supports up to 8 unique CAN Transmit Messages, all of which can be programmed to send any available data to the CAN network.

1.8. 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 MAGIN-CAN Controller on Proprietary B PGNs. However, should a PDU1 message be selected, the MAGIN-CAN 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 MAGIN-CAN Controller supports up to five unique CAN Receive Messages.

1.9. Diagnostic Function Block

There are several types of diagnostics supported by the MAGIN-CAN Signal Controller. Fault detection and reaction is associated with all universal inputs and output drives. In addition to I/O faults, the MAGIN-CAN can also detect/react to power supply over/under voltage measurements, a processor over-temperature, or lost communication events.

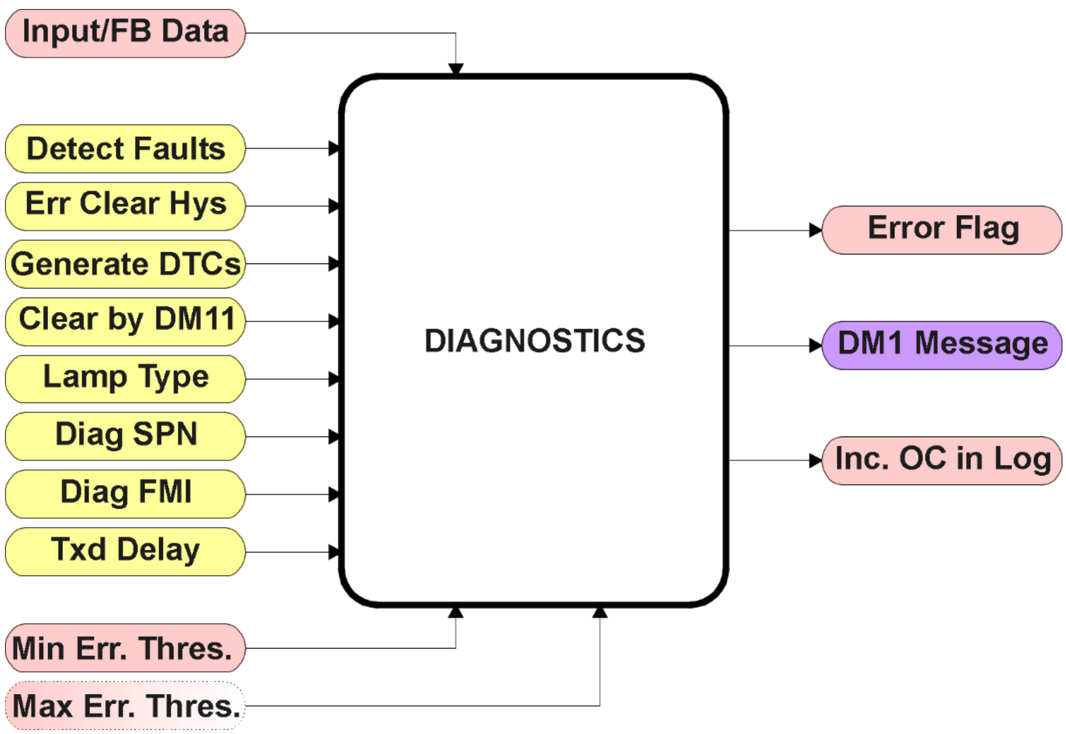


Figure 5 – Diagnostics Function Block

The **"Fault Detection is Enabled"** is the most important setpoint associated with this function block, and it should be selected first. Changing it will result in other setpoints being enabled or disabled as appropriate. When disabled, all diagnostic behaviour associated with the I/O or event in question is ignored.

In most cases, faults can be flagged as either a low or high occurrence. The min/max thresholds for all diagnostics supported by the MAGIN-CAN are listed in Table 11. Bolded values are user configurable setpoints. Some diagnostics react only to a single condition, in which case a N/A is listed in one of the columns.

Function Block	Minimum Threshold	Maximum Threshold
Magnetic Input	**Minimum Error**	**Maximum Error**
Lost Communication	N/A	**Received Message Timeout** (any)

Table 11 – Fault Detect Thresholds

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

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

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

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

In addition to supporting the DM1 message, the MAGIN-CAN Signal Controller also supports

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

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

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

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

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

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

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

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

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

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

Every Diagnostic function block also has associated with it a default FMI. The only setpoint for the user to change the FMI is **"FMI for Event used in DTC,"** even though some Diagnostic function blocks can have both high and low errors as shown in Table 11. In those cases, the FMI in the setpoint reflect that of the low end condition, and the FMI used by the high fault will be determined per Table 12. If the FMI is changed, the OC of the associate error log is automatically reset to zero.

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

Table 12 – Low Fault FMI versus High Fault FMI

If the FMI used is anything other than one of those in Table 12, then both the low and high faults will be assigned the same FMI. This condition should be avoided, as the log will still used different OC for the two types of faults, even though they will be reported the same in the DTC. It is the user's responsibility to make sure this does not happen.

2. Installation Instructions

2.1. Dimensions and Pinout

The MAGIN-CAN Controller is packaged in an ultra-sonically welded plastic housing. The assembly carries an IP67 rating.

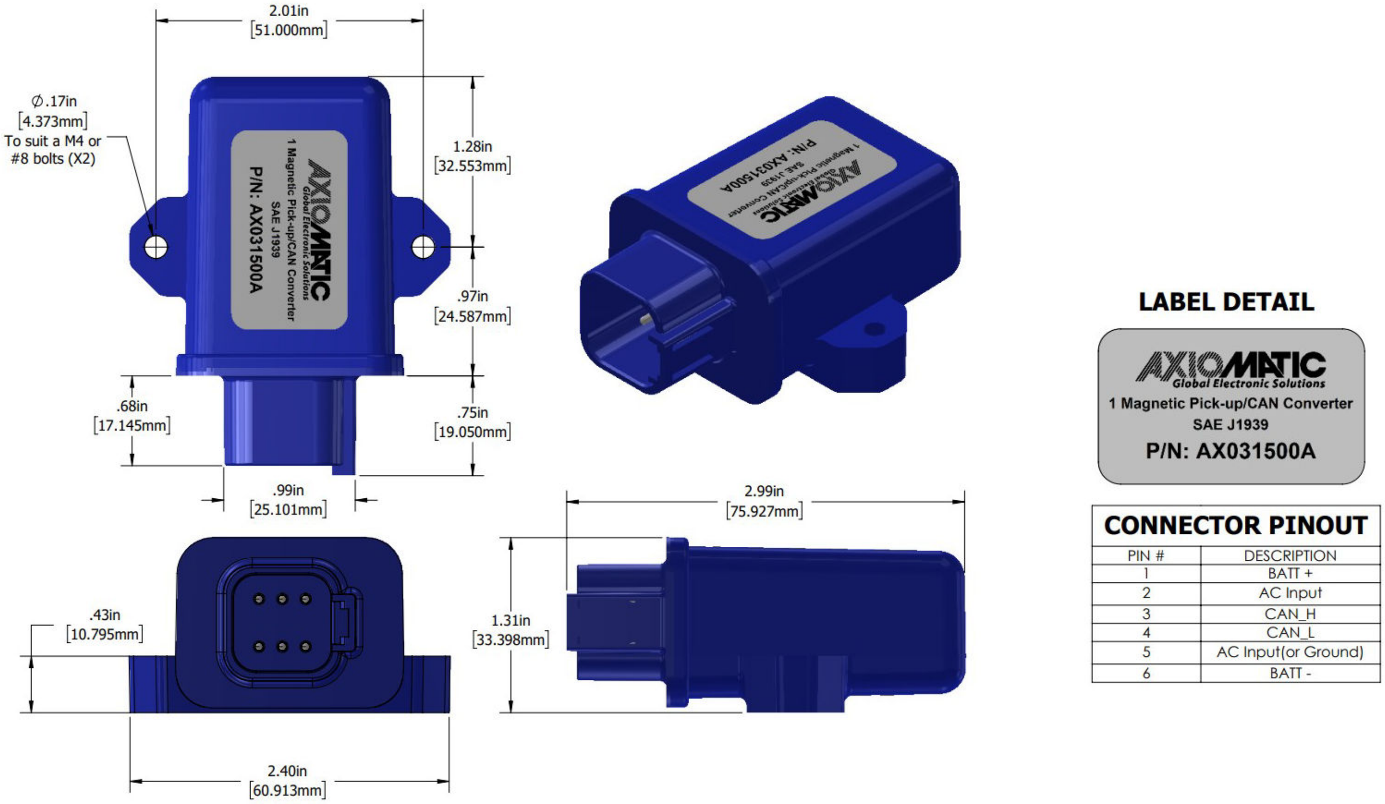


Figure 6 – Housing Dimensions

Pin #	Description
1	BATT+
2	AC Input
3	CAN_H
4	CAN_L
5	AC Input (or Ground)
6	BATT-

Table 13 – 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 Deutsch IPD 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 Deutsch IPD datasheets for usable insulation diameter ranges and other instructions.

Receptacle Contacts	Mating Sockets as appropriate (Refer to www.laddinc.com 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 AX031500A has an auto baud rate feature and is compatible with the following baud rates: 250 kbit/s, 500 kbit/s, 667 kbit/s, and 1 Mbit/s.

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 MAGIN-CAN 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	20, Axiomatic AX031500A, Single Input Controller with CAN
ECU Instance	**0, First Instance**
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 EA will allow 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 MAGIN-CAN 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

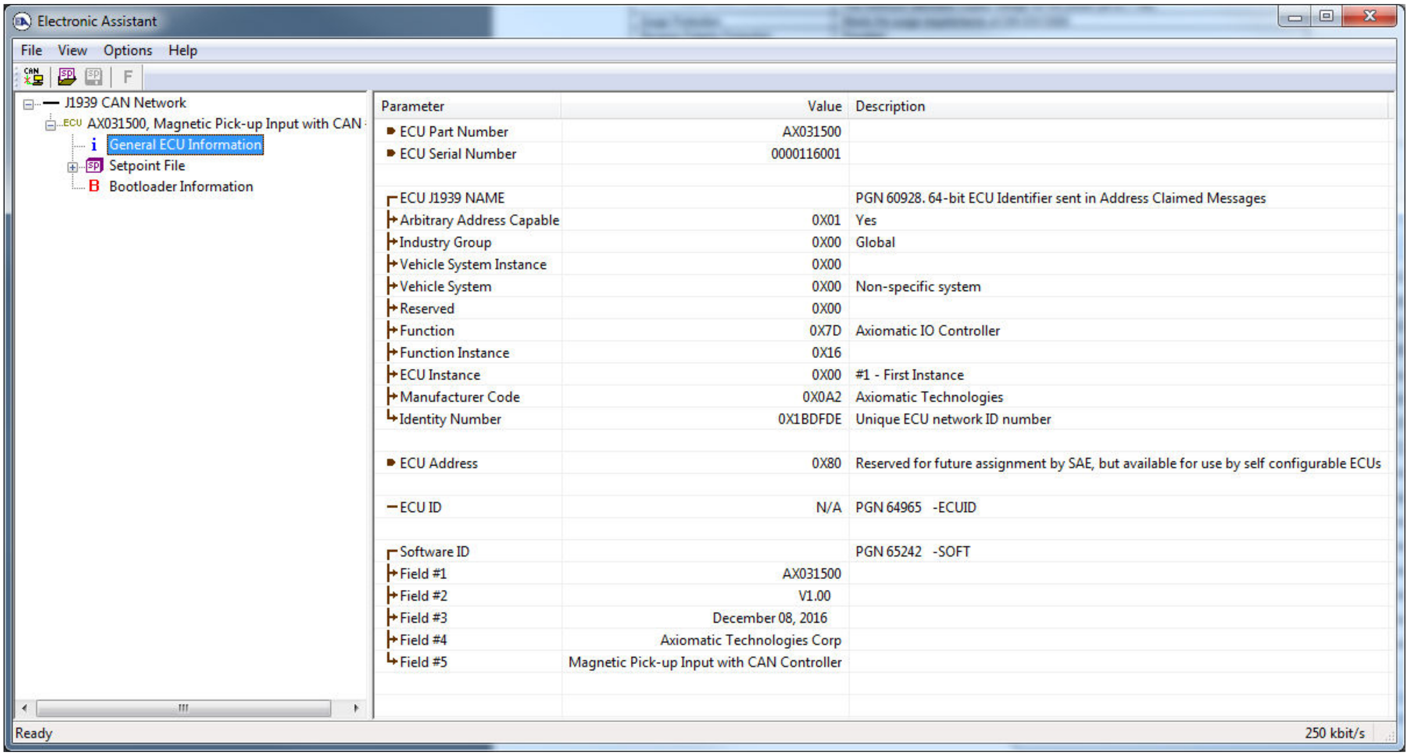
PGN 65242 Software Identification - SOFT
Transmission Repetition Rate: On request
Data Length: Variable
Extended Data Page: 0
Data Page: 0
PDU Format: 254
PDU Specific: 218 PGN Supporting Information:
Default Priority: 6
Parameter Group Number: 65242 (0xFEDA)

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 MAGIN-CAN ECU, Byte 1 is set to 5, and the identification fields are as follows

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

EA shows all this information in “General ECU Information”, as shown below.



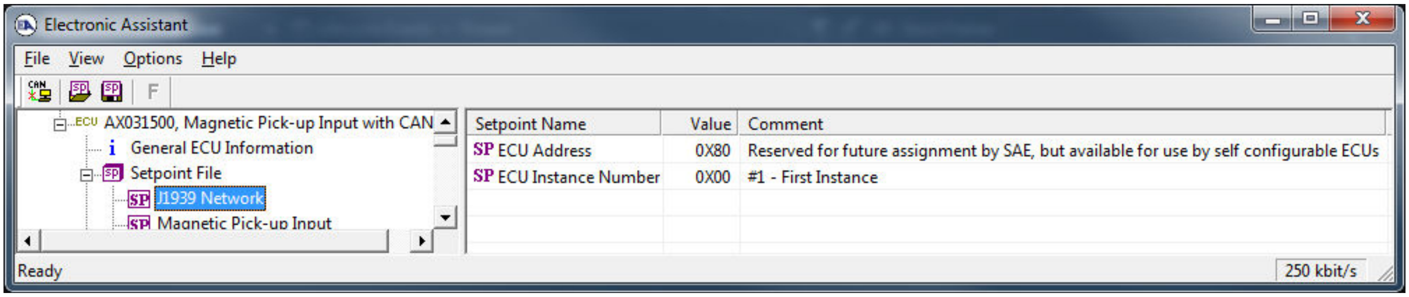
Note: The information provided in the Software ID is available for any J1939 service tool which supports the PGN -SOFT.

4. ECU SETPOINTS ACCESSED WITH 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 MAGIN-CAN, 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.



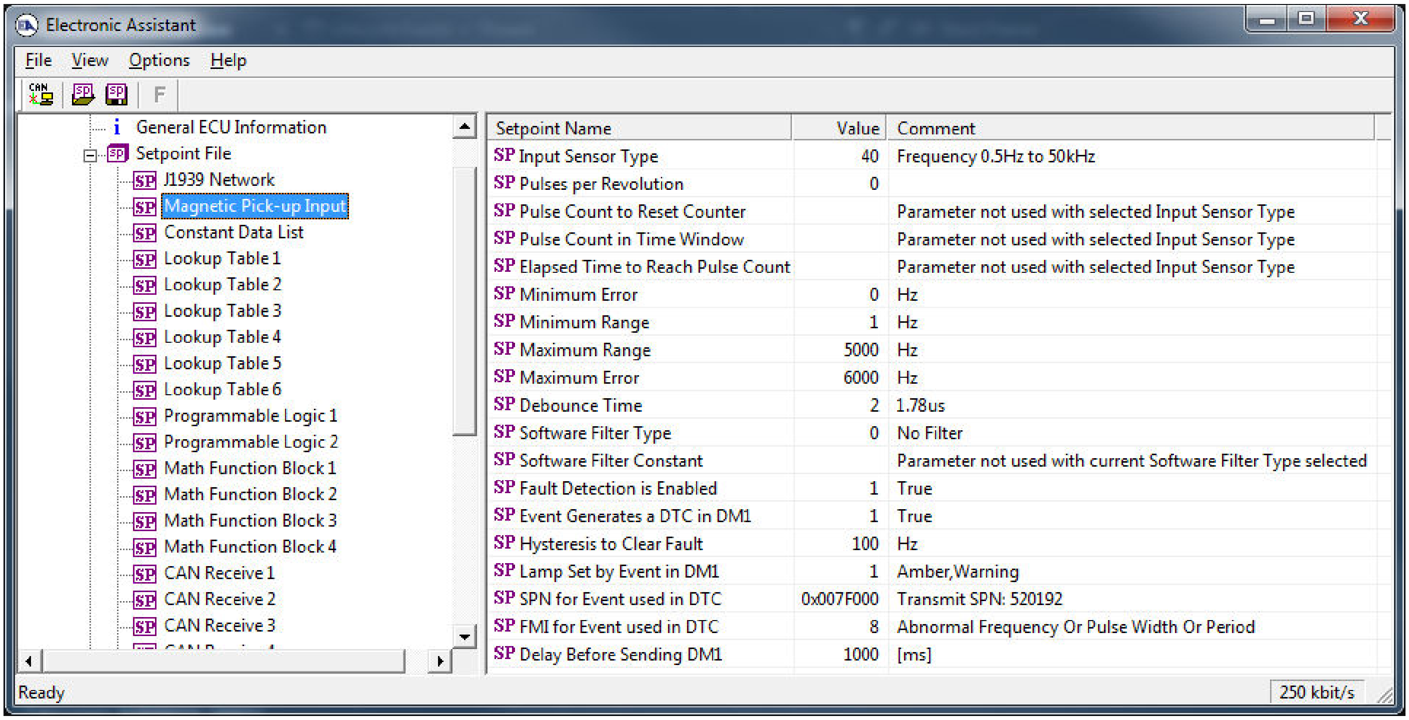
Name	Range	Default	Notes
ECU Instance Number	Drop List	0, #1 – First Instance	Per J1939-81
ECU Address	0 to 253	128 (0x80)	Preferred address for a self-configurable ECU

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 EA after the file is loaded so that only the new NAME and address are showing in the J1939 CAN Network ECU list.

4.2. Magnetic Pick-up Input

The Magnetic Pick-up Input function block is defined in Section 1.2. Please refer to that section for detailed information on how these setpoints are used.



Setpoint Name	Value	Comment
SP Input Sensor Type	40	Frequency 0.5Hz to 50kHz
SP Pulses per Revolution	0	
SP Pulse Count to Reset Counter		Parameter not used with selected Input Sensor Type
SP Pulse Count in Time Window		Parameter not used with selected Input Sensor Type
SP Elapsed Time to Reach Pulse Count		Parameter not used with selected Input Sensor Type
SP Minimum Error	0	Hz
SP Minimum Range	1	Hz
SP Maximum Range	5000	Hz
SP Maximum Error	6000	Hz
SP Debounce Time	2	1.78us
SP Software Filter Type	0	No Filter
SP Software Filter Constant		Parameter not used with current Software Filter Type selected
SP Fault Detection is Enabled	1	True
SP Event Generates a DTC in DM1	1	True
SP Hysteresis to Clear Fault	100	Hz
SP Lamp Set by Event in DM1	1	Amber,Warning
SP SPN for Event used in DTC	0x007F000	Transmit SPN: 520192
SP FMI for Event used in DTC	8	Abnormal Frequency Or Pulse Width Or Period
SP Delay Before Sending DM1	1000	[ms]

Screen Capture of Default Universal Input Setpoints

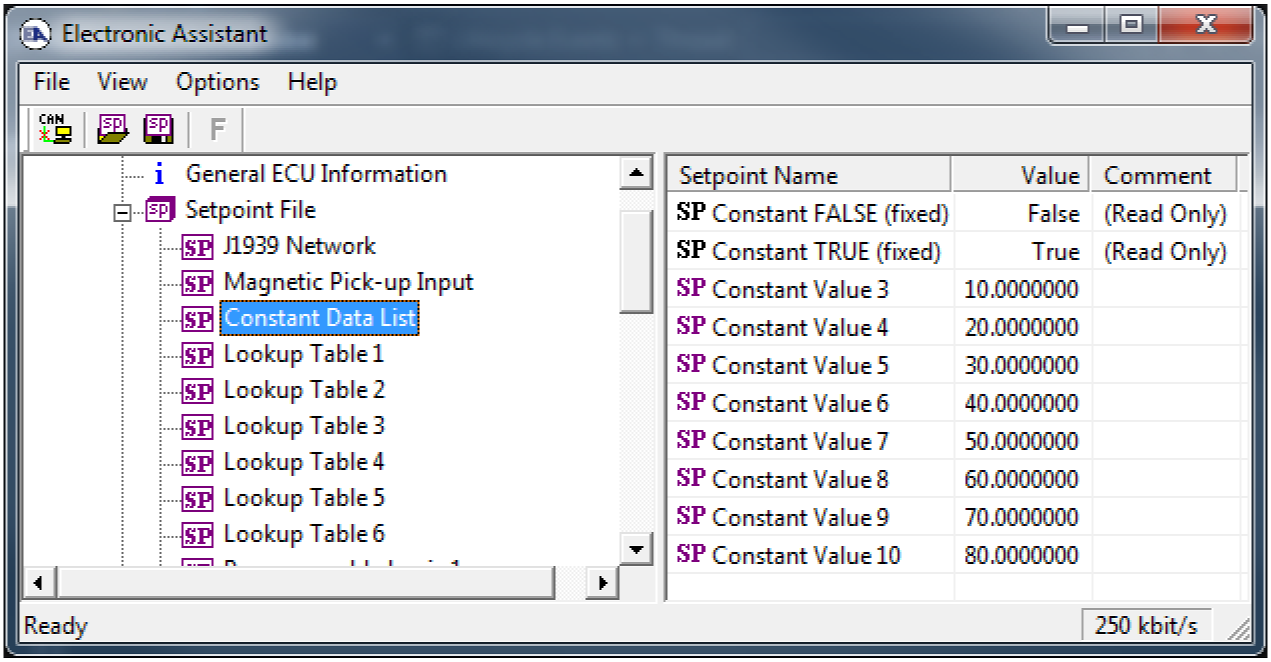
Name	Range	Default	Notes
Input Sensor Type	Drop List	40 – Frequency 0.5Hz to 50kHz	Refer to Section 1.2.1
Pulses per Revolution	0…60000	0	If set to *0*, measurements are taken in Hz. If value is set greater than *0*, measurements are taken in RPM. Refer to Section 1.2.1
Pulse Count to Reset Counter	0…1e+6	1000	Count resets when it reaches the amount and count begins. Refer to Section 1.2.1
Pulse Count in Time Window	0…8.64e+7	500ms	Refer to Section 1.2.1
Elapsed Time to Reach Pulse Count	0…1e+6	1000	Time it takes to reach configured count. Refer to Section 1.2.1
Minimum Error	Depends on Sensor Type	0Hz	Refer to Section 1.2.2
Minimum Range	Depends on Sensor Type	1Hz	Refer to Section 1.2.2
Maximum Range	Depends on Sensor Type	5000Hz	Refer to Section 1.2.2
Maximum Error	Depends on Sensor Type	6000Hz	Refer to Section 1.2.2
Debounce Time	Drop List	0 – *None*	Filter used in 16-bit timer
Software Filter Type	Drop List	0 – *No Filter*	Refer to Section 1.2.3. This function is not used in Counter input types
Software Filter Constant	0 to 60000	1000ms	Refer to Section 1.2.3
Fault Detection is Enabled	Drop List	1 - *True*	Refer to Section 1.9
Event Generates a DTC in DM1	Drop List	1 - *True*	Refer to Section 1.9
Hysteresis to Clear Fault	Depends on Sensor Type	100Hz	Refer to Section 1.9
Lamp Set by Event in DM1	Drop List	1 – *Amber, Warning*	Refer to Section 1.9
SPN for Event used in DTC	0 to 0x1FFFFFFF		Refer to Section 1.9
FMI for Event used in DTC	Drop List	31 – Condition Exists	Refer to Section 1.9
Delay Before Sending DM1	0 to 60000	1000ms	Refer to Section 1.9

4.3. 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

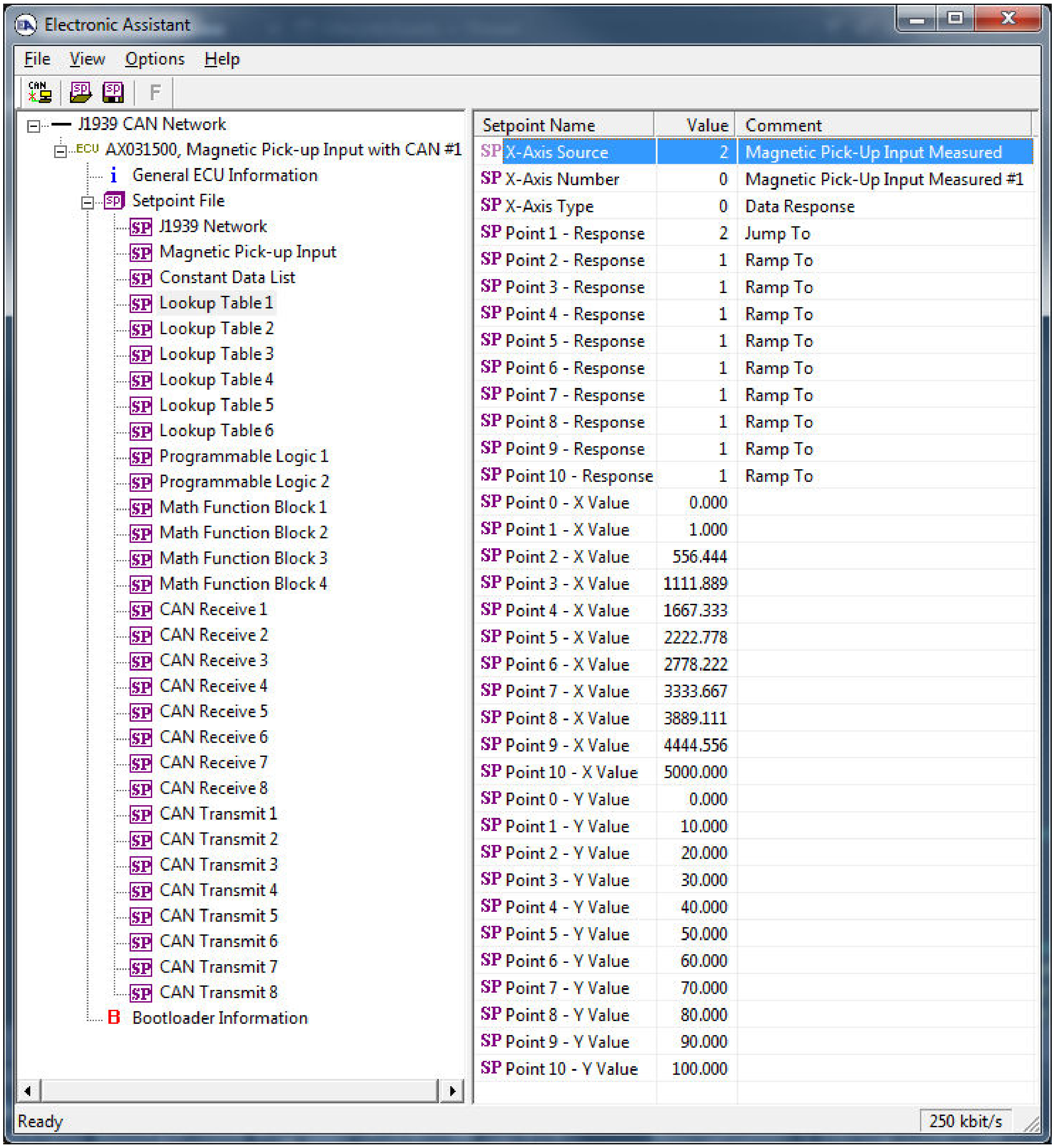
The first two constants are fixed values of 0 (False) and 1 (True) for use in binary logic. The remaining 13 constants are fully user configurable to any value between +/- 1,000,000. The default values are displayed in the screen capture below.



Screen Capture Default Constant Data List Setpoints

4.4. Lookup Table Setpoints

The Lookup Table function block is defined in Section 1.4. 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 1, there is nothing further to define in terms of defaults and ranges beyond that which is described in Section 1.4. Recall, the X-Axis values will be automatically updated if the min/max range of the selected source is changed.



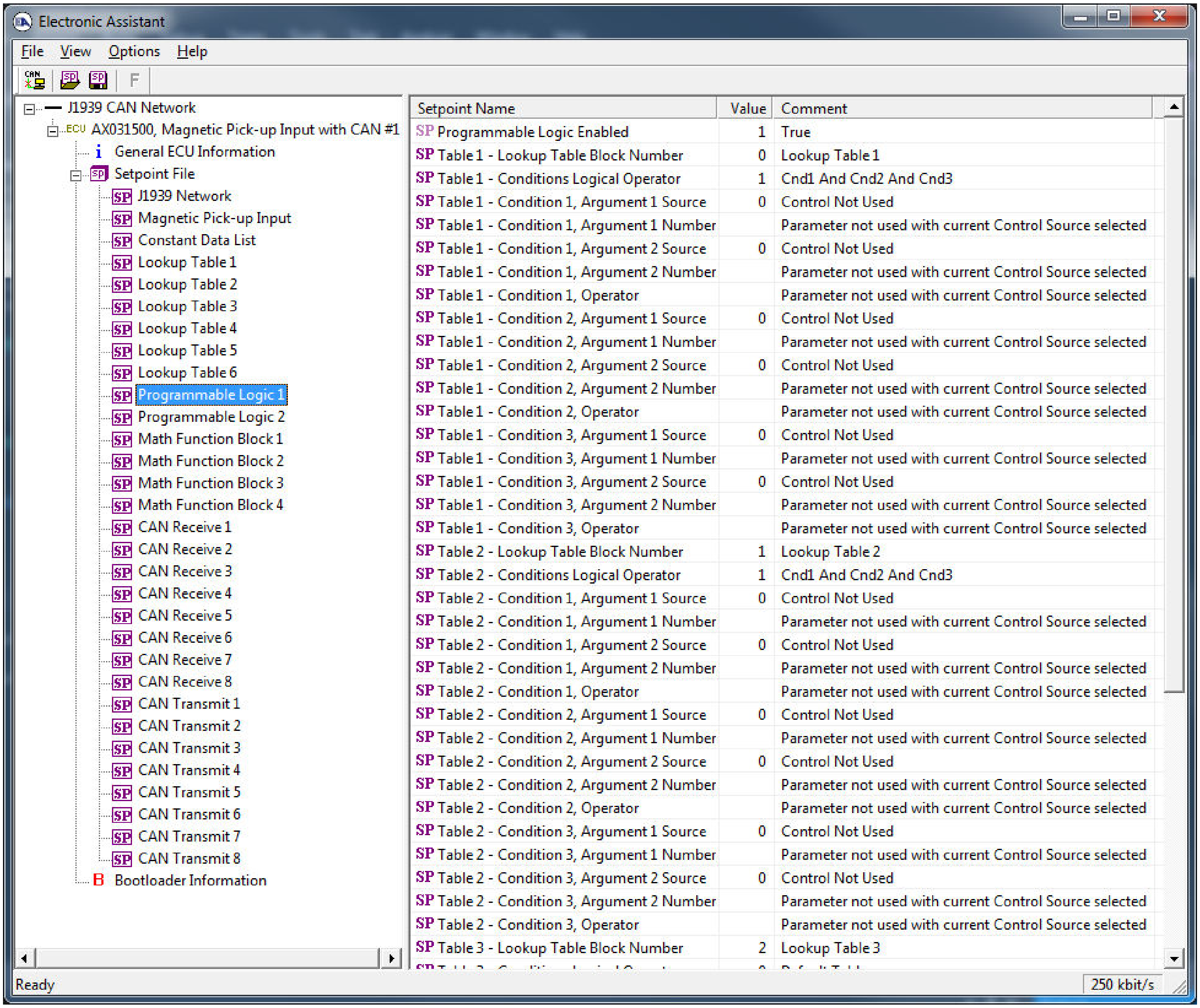
Screen Capture of Example Lookup Table 1 Setpoints

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.

4.5. Programmable Logic Setpoints

The Programmable Logic function block is defined in Section 1.5. 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 1.5. The screen capture below shows how the setpoints referenced in that section appear on EA.



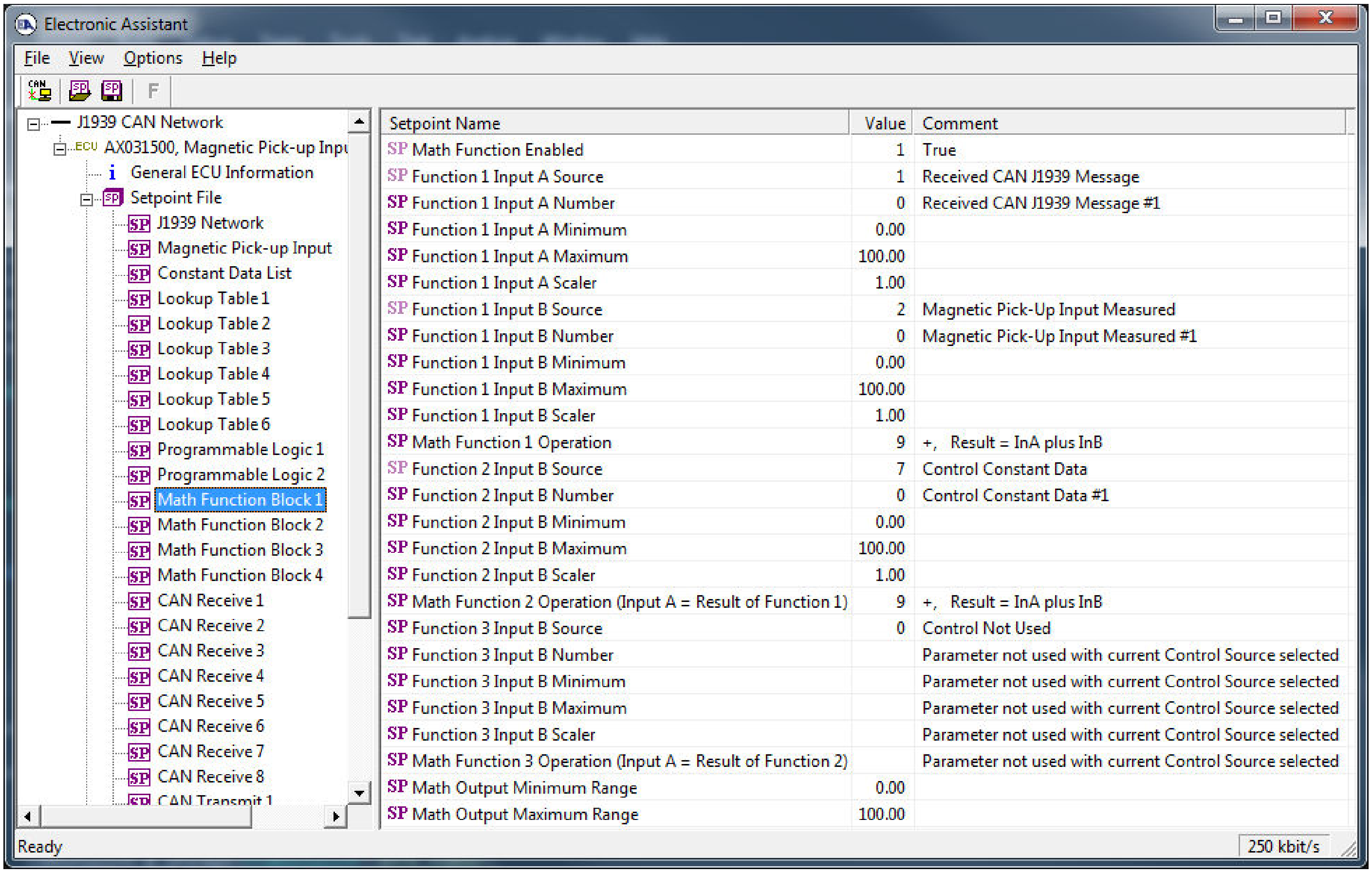
Screen Capture of Default Programmable Logic 1 Setpoints

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.

4.6. Math Function Block Setpoints

The Math 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 Math Function Enabled	1	True
SP Function 1 Input A Source	1	Received CAN J1939 Message
SP Function 1 Input A Number	0	Received CAN J1939 Message #1
SP Function 1 Input A Minimum	0.00	
SP Function 1 Input A Maximum	100.00	
SP Function 1 Input A Scaler	1.00	
SP Function 1 Input B Source	2	Magnetic Pick-Up Input Measured
SP Function 1 Input B Number	0	Magnetic Pick-Up Input Measured #1
SP Function 1 Input B Minimum	0.00	
SP Function 1 Input B Maximum	100.00	
SP Function 1 Input B Scaler	1.00	
SP Math Function 1 Operation	9	+, Result = InA plus InB
SP Function 2 Input B Source	7	Control Constant Data
SP Function 2 Input B Number	0	Control Constant Data #1
SP Function 2 Input B Minimum	0.00	
SP Function 2 Input B Maximum	100.00	
SP Function 2 Input B Scaler	1.00	
SP Math Function 2 Operation (Input A = Result of Function 1)	9	+, Result = InA plus InB
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	100.00	

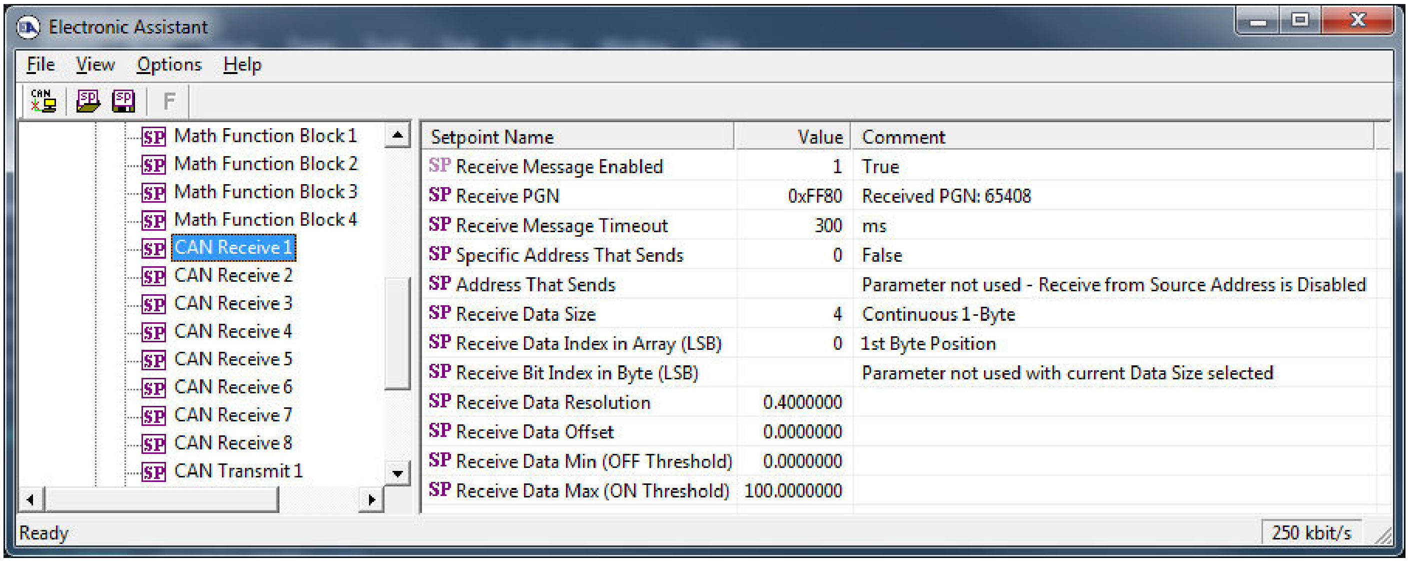
Screen Capture of an Example for Math Function Block

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 1.3
Function 1 Input A Number	Depends on Source	1	Refer to Section 1.3
Function 1 Input A Minimum	-10^6 to 10^6	0	Converts input to percentage before being used in calculation
Function 1 Input A Maximum	-10^6 to 10^6	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 1.6
Function 1 Input B Source	Drop List	0 – *Control Not Used*	Refer to Section 1.3
Function 1 Input B Number	Depends on Source	1	Refer to Section 1.3
Function 1 Input B Minimum	-10^6 to 10^6	0	Converts input to percentage before being used in calculation
Function 1 Input B Maximum	-10^6 to 10^6	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	-10^6 to 10^6	0	Converts input to percentage before being used in calculation
Function 2 Input B Maximum	-10^6 to 10^6	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	-10^6 to 10^6	0	Converts input to percentage before being used in calculation
Function 3 Input B Maximum	-10^6 to 10^6	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	-10^6 to 10^6	0	
Math Output Maximum Range	-10^6 to 10^6	100	

4.7. 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.

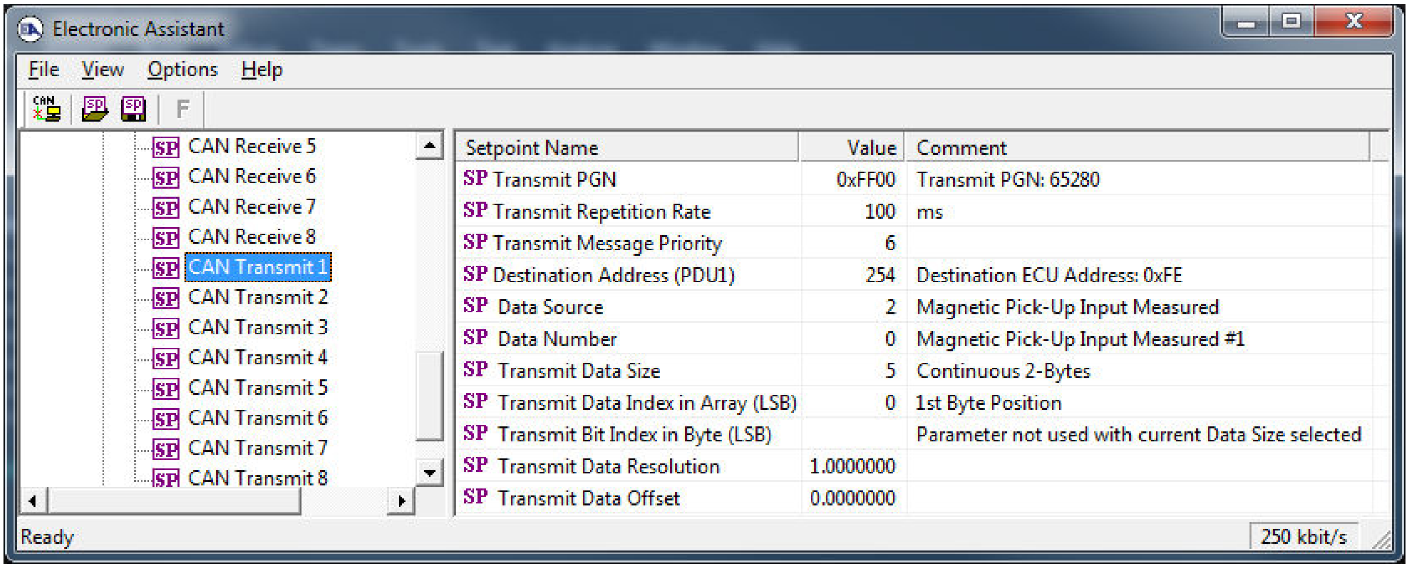


Screen Capture of Default CAN Receive 1 Setpoints

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.

4.8. CAN Transmit Setpoints

The CAN Transmit function block is defined in Section 1.7. Please refer there for detailed information about how all these setpoints are used.



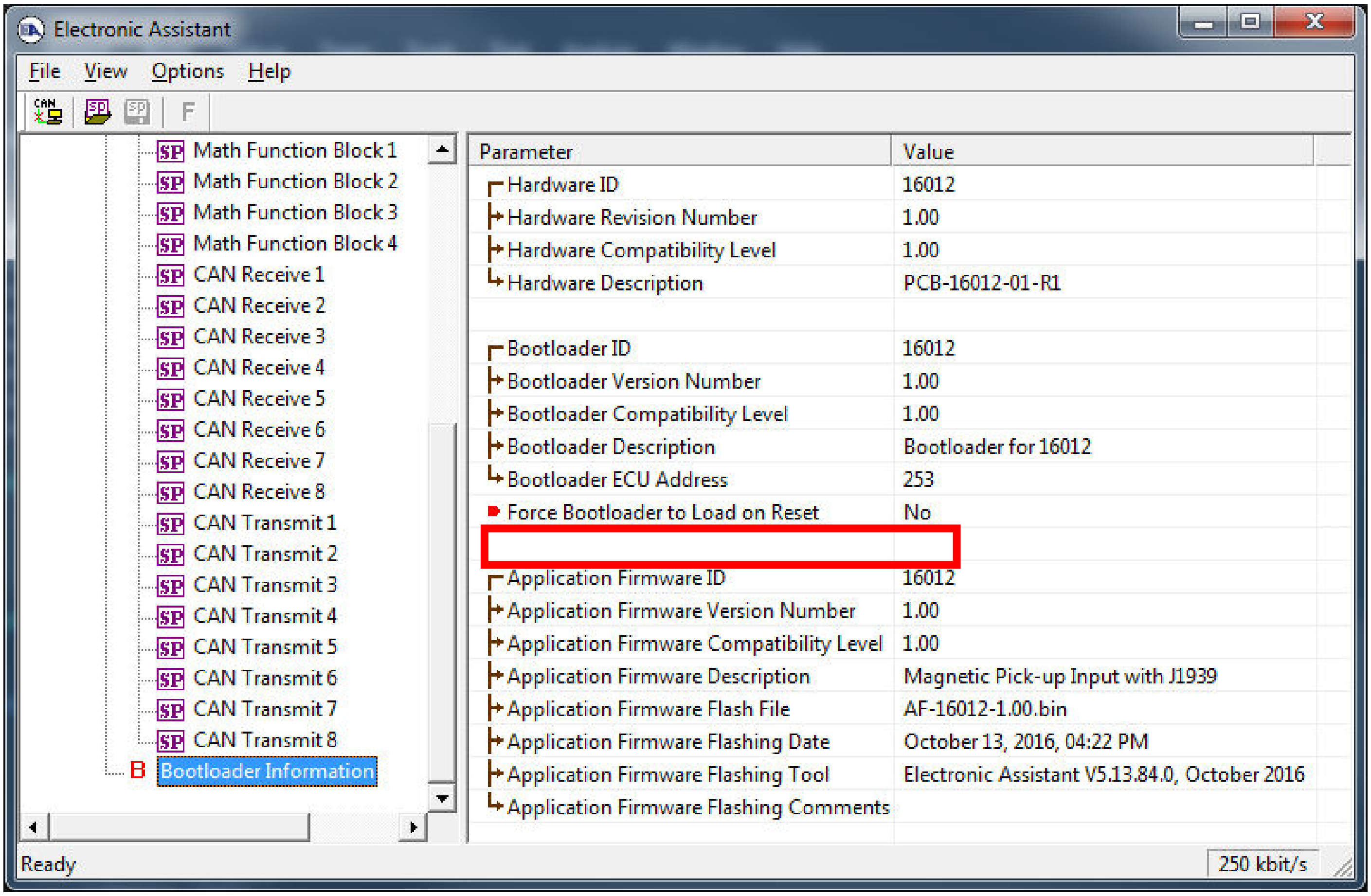
Screen Capture of Default CAN Transmit 1 Setpoints

Name	**Range**	**Default**	**Notes**
Transmit PGN	0…65535	65280 ($FF00)	
Transmit Repetition Rate	0…60,000 ms	0	0ms disables transmit
Transmit Message Priority	0…7	6	Proprietary B Priority
Destination Address (for PDU1)	0…255	254 (0xFE, Null Address)	Not used by default
Transmit Data Source	Drop List	Input Measured	Refer to Section 1.3
Transmit Data Number	Per Source	0, Input Measured #1	Refer to Section 1.3
Transmit Data Size	Drop List	Continuous 1-Byte	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)	0 to 8-DataSize	0, First Byte Position	
Transmit Bit Index in Byte (LSB)	0 to 8-BitSize	Not Used by Default	Only used with Bit Data Types
Transmit Data Resolution	-10^6 to 10^6	1.00	
Transmit Data Offset	-10^4 to 10^4	0.00	

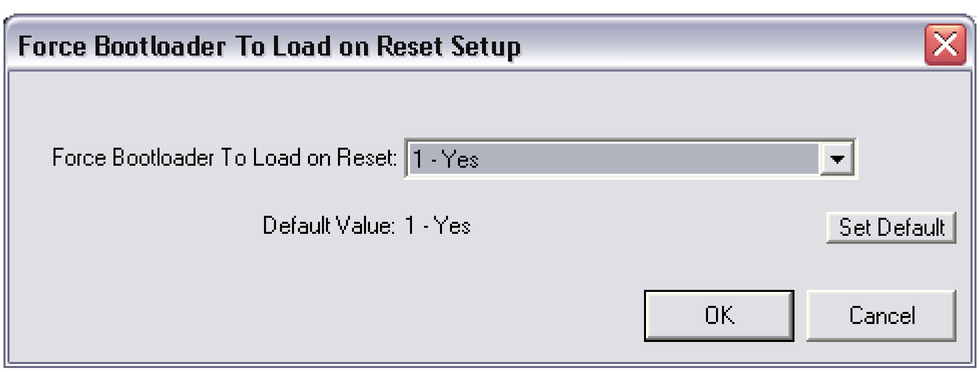
5. REFLASHING OVER CAN WITH EA BOOTLOADER

The AX031500A 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.

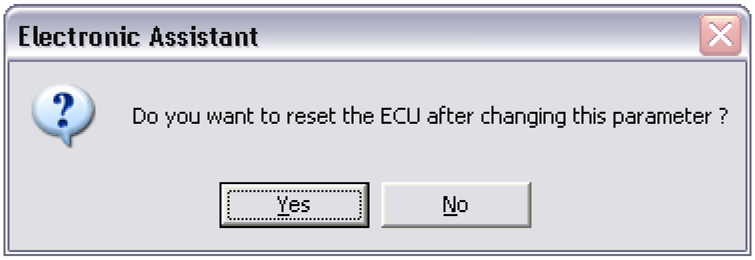
1. When EA first connects to the ECU, the **Bootloader Information** section will display the following information.



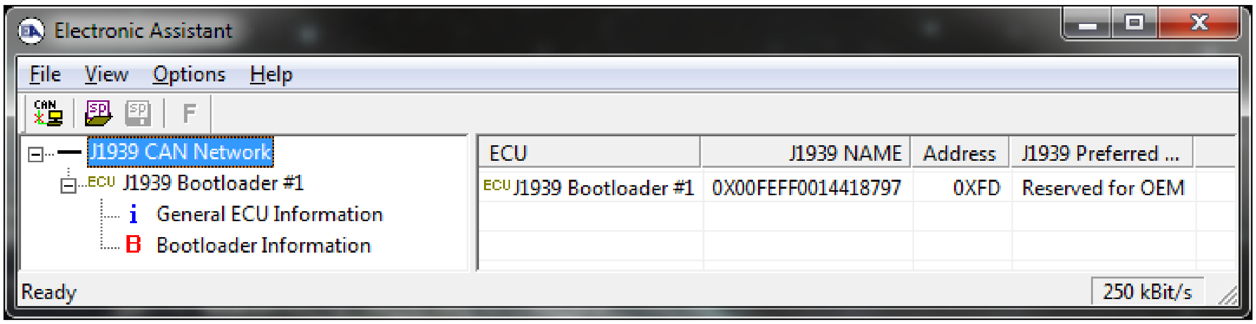
2. To use the bootloader to upgrade the firmware running on the ECU, change the variable **"Force Bootloader To Load on Reset"** to *Yes.*

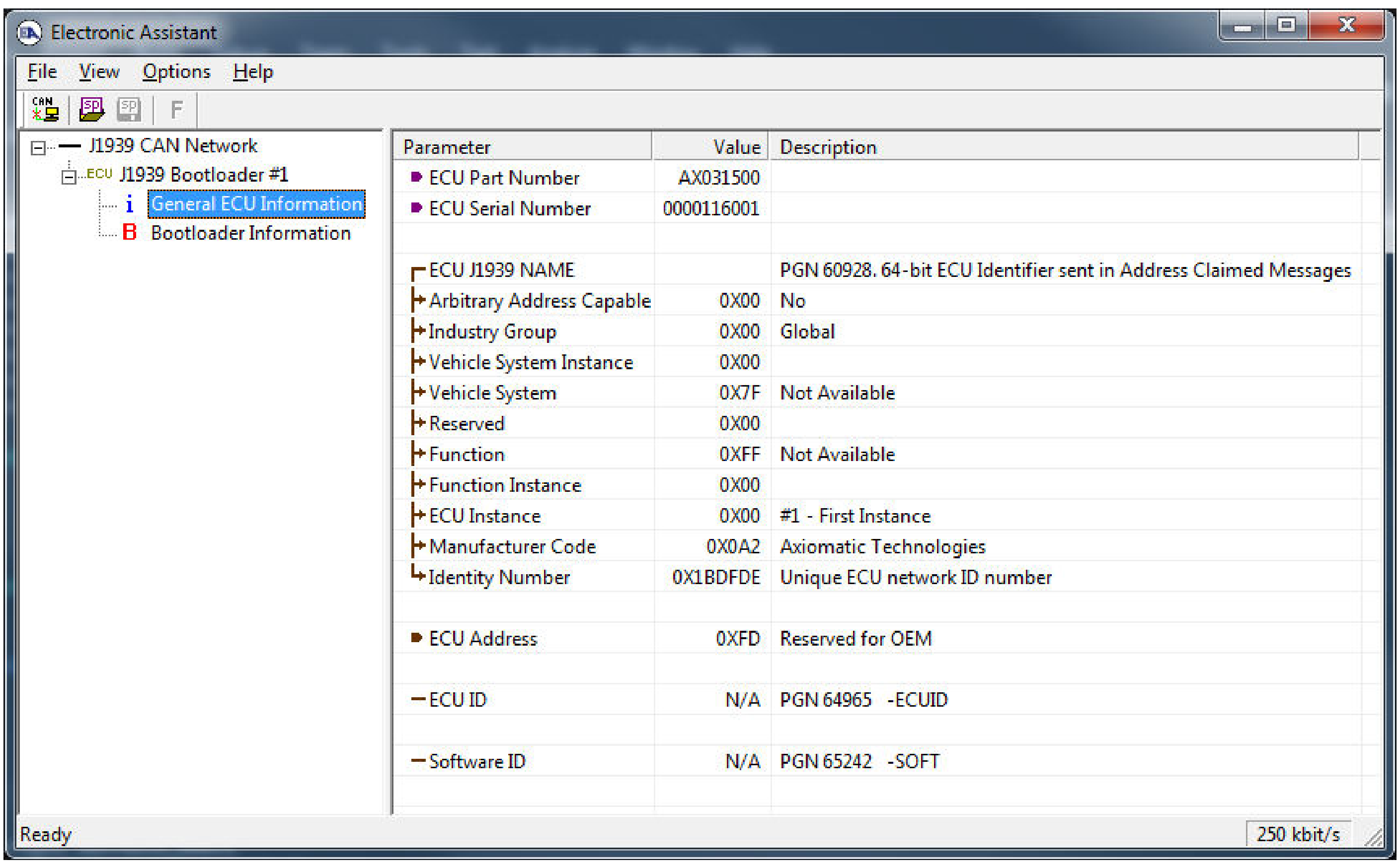


3. When the prompt box asks if you want to reset the ECU, select *Yes*.



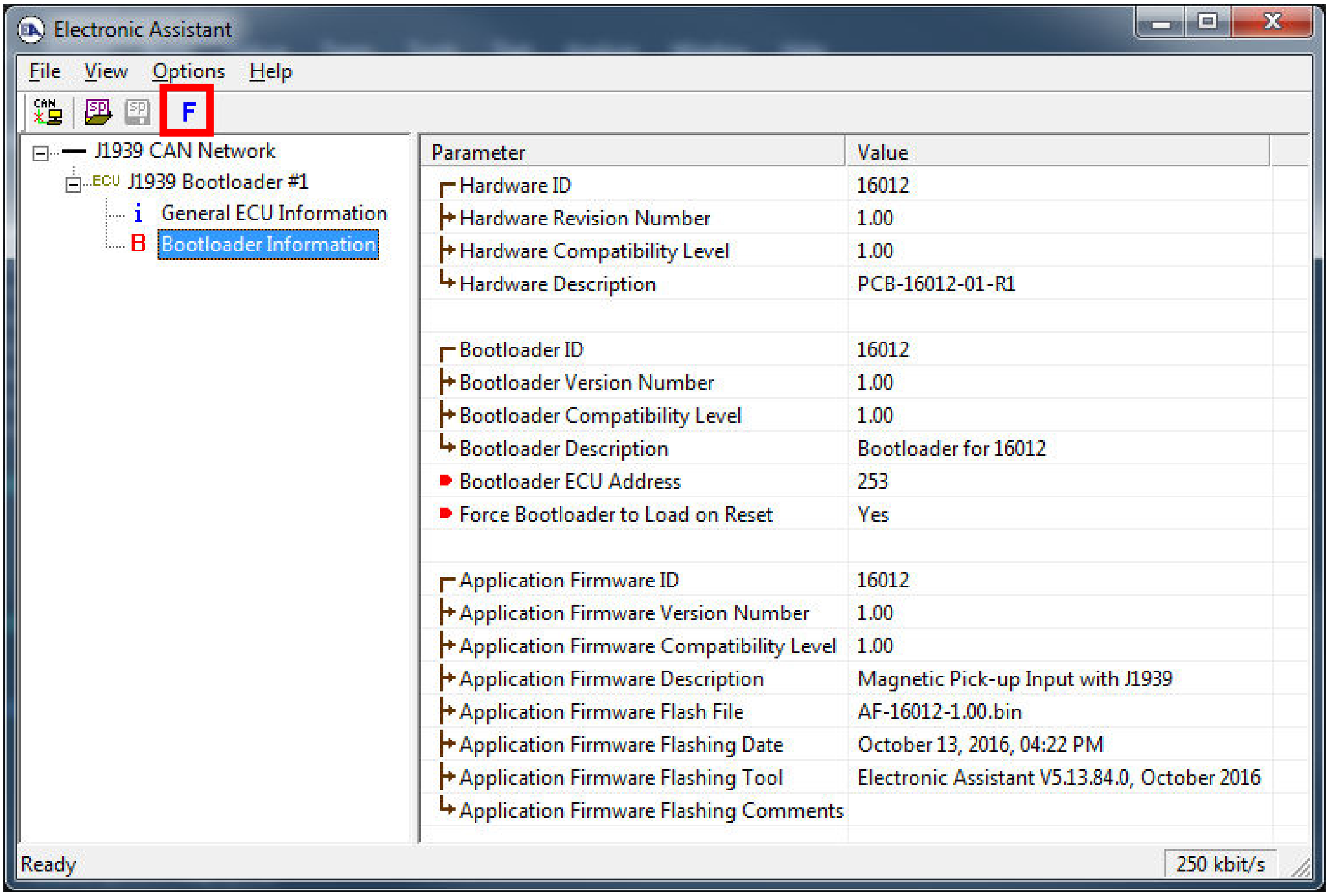
4. Upon reset, the ECU will no longer show up on the J1939 network as an AX031500A but rather as **J1939 Bootloader #1**.





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.

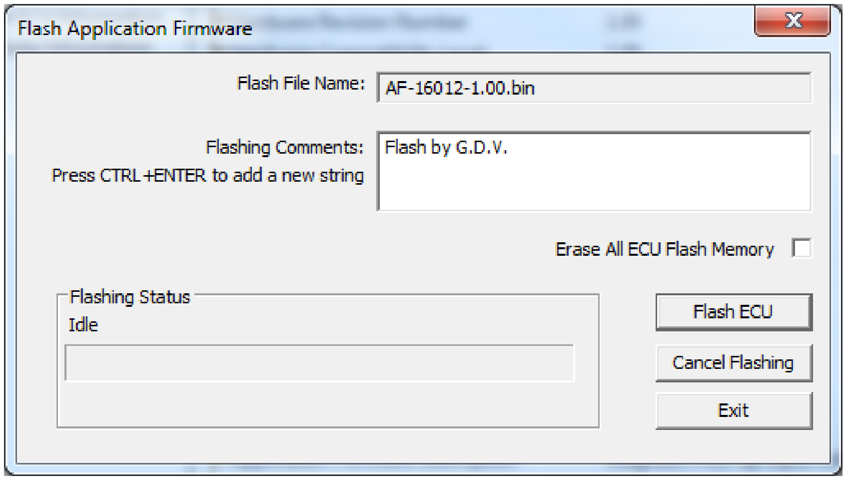
5. When the **Bootloader Information** section is selected, the same information is shown as when it was running the AX031500A firmware, but in this case the **F**lashing feature has been enabled.



6. Select the **F**lashing button and navigate to where you had saved the **AF-16012-x.yy.bin** file sent from Axiomatic. (Note: only binary (.bin) files can be flashed using the EA tool)

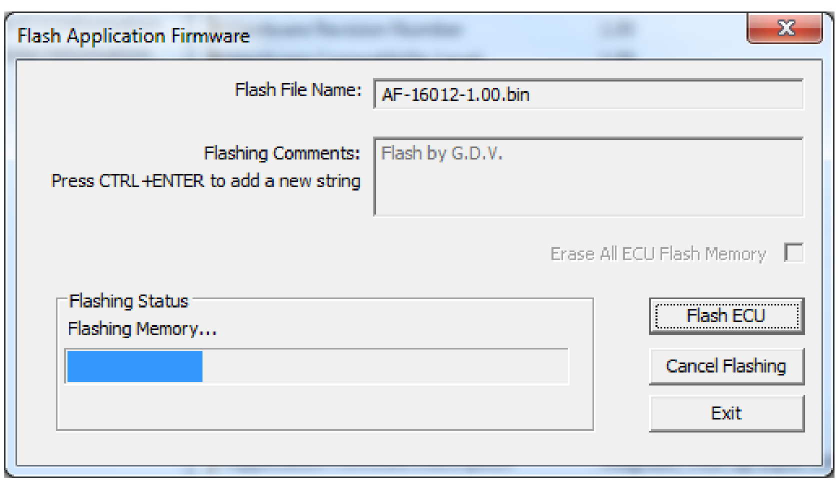
7. Once the Flash Application Firmware window opens, you can enter comments such as “Firmware upgraded by [Name]” if you so desire. This is not required, and you can leave the field blank if you do not want to use it.

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



> WARNING: Do not check the “Erase All ECU Flash Memory” box unless instructed to do so by your Axiomatic contact. Selecting this will erased ALL data stored in non-volatile flash. 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 AX031500A application will start running, and the ECU will be identified as such by EA. Otherwise, the next time the ECU is power-cycled, the AX031500A 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 Limitations & Return Materials Process as described on https://www.axiomatic.com/service/.

Power

Power Supply Input	12 V, 24 V or 48 VDC nominal; 8 to 60 VDC range The minimum allowable supply voltage for the power pin is 7 VDC.
Quiescent Current Draw	9.1 mA @ 24 VDC; 16 mA @ 12 VDC typical
Surge Protection	Meets the surge requirements of DIN EN13309
Reverse Polarity Protection	Provided

Input

Input	1 magnetic pick-up sensor input configurable as follows. • Frequency (0.5 Hz to 50 kHz) • Counter (Pulse count reset) • Counter (Time window) • Counter (Pulse window) Amplitude: 0.1 VAC (RMS) to 48 VAC (RMS)
Sampling Time	The input is tied to a 16-bit timer.

General Specifications

Microcontroller	STM32F103CBT7, 32-bit, 128 KB Flash Program Memory
CAN Port	1 CAN (SAE J1939) Supported Baud Rates: 250 kbit/s, 500 kbit/s, 667 kbit/s, and 1 Mbit/s with auto-baud-rate detection CANopen® model - P/N: AX031501A
Network Termination	According to the CAN standard, it is necessary to terminate the network with external termination resistors. The resistors are 120 Ω, 0.25 W minimum, metal film or similar type. They should be placed between CAN_H and CAN_L terminals at both ends of the network.
User Interface	Axiomatic Electronic Assistant KIT - P/N: **AX070502** or **AX070506K**
Compliance	CE / UKCA marking RoHS
Vibration	MIL-STD-202G, method 204D, test condition C 10 g peak (Sine) MIL-STD-202G, method 214A, test condition I/B 7.86 Grms peak (Random)
Shock	MIL-STD-202G, method 213B, test condition A 50 g peak
Operating Temperature	-40 to 85°C (-40 to 185°F)
Storage Temperature	-50 to 125°C (-58 to 257°F)
Protection Rating	IP67
Enclosure and Dimensions	Plastic Enclosure, Nylon 6-6 with 30% glass fill Integral connector Flammability Rating: UL 94V-0 Refer to Figure 1.0 - dimensional drawing.
Weight	0.10 lb. (0.045 kg)
Electrical Connections	6-pin connector (equivalent to TE Deutsch P/N: DT04-6P)
Mating Plug Kit	A mating plug kit is available under Axiomatic P/N: **AX070119** (includes 1x plug DT06-6S, 6x contacts 0462-201-16141, and 1x wedgelock W6S)

7. VERSION HISTORY

Version	Date	Author	Modifications
1.0.0	April 27, 2025	Dmytro Tsebrii	Initial draft
-	June 12, 2025	A Wilkins	Marketing review
1.0.1	July 22, 2025	M Ejaz	Updated input voltage rating Updated dimensional drawing

OUR PRODUCTS

AC/DC Power Supplies

Actuator Controls/Interfaces

Automotive Ethernet Interfaces

Battery Chargers

CAN Controls, Routers, Repeaters

CAN/WiFi, CAN/Bluetooth, Routers

Current/Voltage/PWM Converters

DC/DC Power Converters

Engine Temperature Scanners

Ethernet/CAN Converters, Gateways, Switches

Fan Drive Controllers

Gateways, CAN/Modbus, RS-232

Gyroscopes, Inclinometers

Hydraulic Valve Controllers

Inclinometers, Triaxial

I/O Controls

LVDT Signal Converters

Machine Controls

Modbus, RS-422, RS-485 Controls

Motor Controls, Inverters

Power Supplies, DC/DC, AC/DC

PWM Signal Converters/Isolators

Resolver Signal Conditioners

Service Tools

Signal Conditioners, Converters

Strain Gauge CAN Controls

Surge Suppressors

OUR COMPANY

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

QUALITY DESIGN AND MANUFACTURING

We have an ISO9001:2015 registered design/manufacturing facility in Canada.

WARRANTY, APPLICATION APPROVALS/LIMITATIONS

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

COMPLIANCE

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

SAFE USE

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

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

SERVICE

All products to be returned to Axiomatic require a Return Materials Authorization Number (RMA#) from rma@axiomatic.com. Please provide the following information when requesting an RMA number:

- Serial number, part number
- Runtime hours, description of problem
- Wiring set up diagram, application and other comments as needed

DISPOSAL

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

CONTACTS

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

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