USER MANUAL UMAX140620
Version 1.0.1

LIN/CAN Converter

SAE J1939

USER MANUAL

P/N: AX140620

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

ACRONYMS

CAN	Controller Area Network
CANopen®	CANopen® is a registered community trademark of CAN in Automation e.V.
CAN-ID	CAN 11-bit Identifier
COB	Communication Object
CTRL	Control
EDS	Electronic Data Sheet
EMCY	Emergency
LSB	Least Significant Byte (or Bit)
LSS	Layer Settling Service
LUT	Lookup Table
MSB	Most Significant Byte (or Bit)
NMT	Network Management
PID	Proportional-Integral-Derivative Control
RO	Read Only Object
RPDO	Received Process Data Object
RW	Read/Write Object
SDO	Service Data Object
TPDO	Transmitted Process Data Object
WO	Write Only Object

REFERENCES

[DS-301] CiA DS-301 V4.1 – CANopen® Application Layer and Communication Profile. CAN in Automation 2005

[DS-305] CiA DS-305 V2.0 – Layer Setting Service (LSS) and Protocols. CAN in Automation 2006

These documents are available from the CAN in Automation e.V. website: http://www.can-cia.org/.

TABLE OF CONTENTS

1. OVERVIEW OF CONTROLLER

1.1. Description of I/O Controller

The converter is designed to translate application signals between LIN 2.2 and J1939 networks.

The converter accepts power supply voltages from 9 to 36 VDC. It can run in LIN master or slave mode at different baud rate from 2.4…20 kbit/s. The J1939 CAN network can operate at standard 250 kbit/s and 500 kbit/s baud rates and non-standard 125 kbit/s, 667 kbit/s and 1Mbit/s baud rates. The required baud rate is detected automatically upon connection to the CAN network. The converter can be configured through a set of configuration parameters to fit the user specific application requirements.

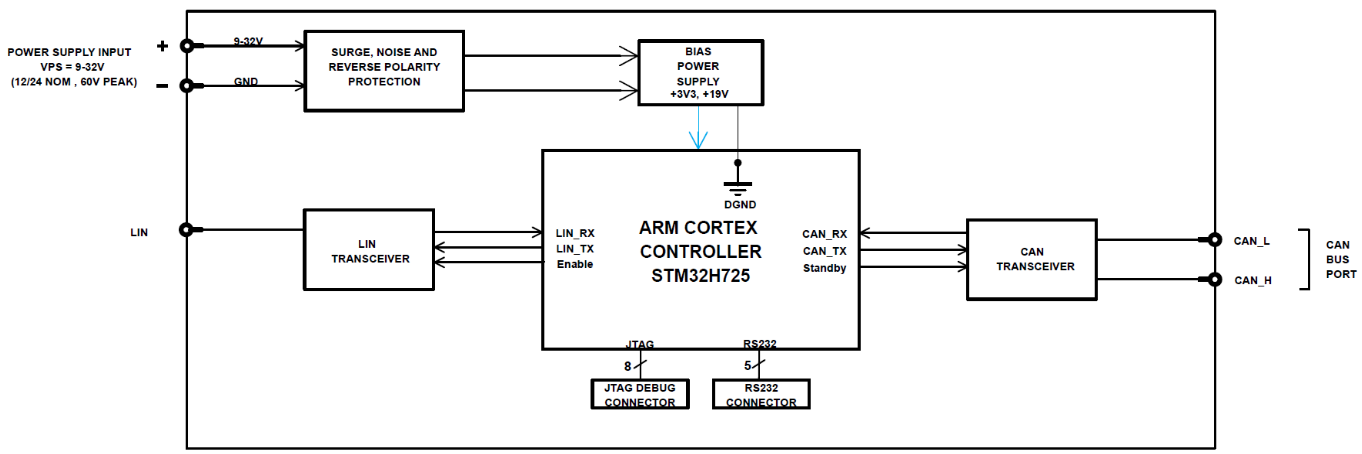


Figure 1 – Hardware Functional Block Diagram

1.1.1. Description of LIN Interface

The LIN interface is compliant with the LIN Specification Package, Revision 2.2A, December 31, 2010. The following parts of this standard specification package were implemented:

Table 1 – LIN Interface Information

ISO/OSI Network Model Layer	LIN Specification Package Document
Physical	Physical Layer Specification.
Data Link	Protocol Specification.
Network	Not Implemented.
Transport	N/A in LIN.
Session	N/A in LIN.
Presentation	N/A in LIN.
Application	Application Program Interface Specification. Master and Slave nodes are supported. The user can configure: 130 signals, 25 unconditional frames, 1 event triggered and 1 sporadic frame. One main schedule table and one collision schedule table are also available for master nodes.

1.2. Available Control Sources

Many of the function blocks have selectable input signals, which are determined with Source setpoints. This setpoint uniquely select how the I/O of the various function blocks are linked together. Available Source options are listed in Table 2. All sources are available for all blocks. Though input Sources are freely selectable, it must be remembered that not all options would make sense in all cases, and it is up to the user to configure the controller in a logical and functional manner.

Table 2 – Control Source Options

Value	Meaning	Range
0	Control Source Not Used (Ignored)	0
1	Can Receive	1-25
2	LIN Signal	1-130
3	Constant Data	1-15
4	Binary Function Output	1-16
5	DTC React	1-16
6	Processor Temperature Measured	1-255
7	Power Supply Measured	1-255
8	Receive Message Timeout	1
9	LIN Slave Mode Response Error	1

1.2.1. Function Block Signals

The converter's function blocks can communicate with each other through signal inputs and outputs. Each signal input can be connected to any signal output using an appropriate configuration parameter.

When a signal input is connected to a signal output, data from the signal output of one function block is available on the signal input of another function block.

Depending on the signal type, the function block signal output can be either Discrete or Continuous. The function block signal input, receiving the output signal, can be: Undefined, Discrete or Continuous.

1.2.1.1. Undefined Signal

The *Undefined* signal type is used to present a no-signal condition in signal data or to specify that the signal input is not connected (not used).

1.2.1.2. Discrete Signal

The *Discrete* signal type is used to present a discrete signal that has a finite number of states in signal data or to specify that the signal input or output is communicating this type of signals.

The discrete signals are stored in four-byte unsigned integer variables that can present any state value in the 0…0xFFFFFFFF range.

1.2.1.3. Continuous Signal

The *Continuous* signal type presents continuous signals, usually physical parameters, in signal data or as a signal input or output type.

The continuous signals are stored in floating point variables. They are not normalized and present data in the appropriate physical units. The user can do simple scaling of the continuous signal data by changing *Scale (Resolution)* and *Offset* configuration parameters in the appropriate function blocks.

1.2.1.4. Discrete to Continuous Conversion

A *Discrete* signal is converted into a positive *Continuous* signal of the same value.

1.2.1.5. Continuous to Discrete Conversion

A positive *Continuous* signal is converted into the same value *Discrete* signal. A fractional part of the *Continuous* signal is truncated. If the *Continuous* signal value is above the maximum *Discrete* signal value, the resulted *Discrete* signal value will saturate at the maximum *Discrete* signal value: 0xFFFFFFFF.

All negative *Continuous* signals are converted into zero value *Discrete* signals.

1.2.1.6. Undefined Signal Conversion

An *Undefined* signal is not converted into a specific discrete or continuous signal value. It presents a no-signal condition on both: *Discrete* and *Continuous* signal inputs and outputs. The value of an undefined signal is not defined unless a default signal value configuration parameter is used in a function block. In this case, the configuration parameter value is used as a signal value when the signal is not defined, see Section 1.5 for information on Binary Function blocks.

1.3. LIN Interface

The converter LIN interface is defined by: *LIN Signal*, *LIN Unconditional Frame*, *Event Triggered Frame*, *Sporadic Frame*, *Main Schedule Table, Collision Schedule Table* and *LIN Common* function blocks.

LIN signals are sent and received through *LIN Signal* function blocks. Other function blocks are used to define the LIN network communication.

The function blocks will be presented in the order they appear in the CANopen® Object Directory.

1.3.1. LIN Common

LIN Common function block does not have any signal inputs or outputs.

Figure 2 – LIN Common Function Block

It defines the high-level LIN bus configuration parameters. See the following table.

Table 3 – LIN Common Function Block Configuration Parameters

Name	Default Value	Range	Units	Description
Node Type	Master	{Undefined, Master, Slave}	–	LIN node type. The "Undefined" value is used to switch off the node, for example during debugging.
Baud Rate	19200	[2400...20000]	bit/s	LIN bus baud rate. The minimum baud rate value is limited by the converter transceiver hardware.
Tick Time	10	[1...10000]	ms	Tick time, if Node Type is "Master".

1.3.2. LIN Signal

LIN Signal function blocks are used to specify input and output signals on the LIN bus. There are *130 LIN Signal* function blocks available to the user. Each *LIN Signal* function block has one signal input and one signal output for interfacing with other function blocks.

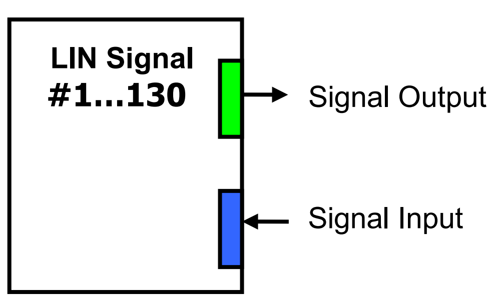


Figure 3 – LIN Signal Function Block

Configuration parameters of the *LIN Signal* function block are presented below:

Table 4 – LIN Signal Function Block Configuration Parameters

Name	Default Value[1]	Range	Units	Description
LIN Signal Type	Scalar	{Undefined, Scalar, Byte Array}	–	Type of the LIN signal.
Input Signal Source	0 - No Source	Any signal output of any function block or “Not Connected”	–	Input signal source if LIN signal is output (data is sent on the LIN bus).
Input Signal Number	0	Dependent on 2191h source	-	Input Signal number
Output Signal Autoreset Time	0	[1…10000]	ms	Autoreset time if LIN signal is input (data is received from the LIN bus). If 0 – Autoreset is disabled.
Size	8	[1…64]	Bit	Signal size.
Encoding Type	Physical Value	{Undefined, Logical Value, Physical Value, BCD Value, ASCII Value}	–	Type of the LIN signal content.
Min Value	0	[0…0xFFFFFFFF]	–	Min value of the signal code when the Encoding Type is “Logical Value” or “Physical Value”.
Max Value	0xFE	[0…0xFFFFFFFF]	–	Max value of the signal code when the Encoding Type is “Logical Value” or “Physical Value”.
Scale	1	Any value	Signal Units/Bit	Signal scale when the Encoding Type is “Physical Value”.
Offset	0	Any value	Signal Units	Signal offset when the Encoding Type is “Physical Value”.
Init Value Scalar	0x1	[0…0xFFFF]	–	Initial signal value when LIN Signal Type is Scalar.
Init Value Byte Array [0]	0	[0…0xFF]	–	Initial signal value of the 1-st byte when LIN Signal [xy] Type is “Byte Array”.
Init Value Byte Array [1]	0	[0…0xFF]	–	Initial signal value of the 2-nd byte when LIN Signal [xy] Type is “Byte Array”.
…	…	…	…	…
Init Value Byte Array [7]	0	[0…0xFF]	–	Initial signal value of the 8-th byte when LIN Signal [xy] Type is “Byte Array”.

Encoding Type configuration parameter defines the function block signal input and output type the following way:

Table 5 – LIN Signal Encoding Type

Encoding Type	Function Block Signal Type
Undefined	Undefined
Logical Value	Discrete
Physical Value	Continuous
BCD Value	Discrete
ASCII Value	Discrete

1.3.2.1. Receiving LIN Signals

When the *LIN Signal* function block receives signals from the LIN bus, they are converted to the function block output signal the following way:

- Logical signals are received only when they are within the [*MinValue; MaxValue*] range. No conversion is performed;
- BCD signals are received unconditionally. No conversion is performed;
- ASCII signals are masked with 0xFF value. This way, only the least significant byte of the input signal is received and passed to the output;
- Physical signals are received only when they are within the [*MinValue; MaxValue*] range and then converted to the output signal value using *Scale* and *Offset* configuration parameters:

$$LINOutputSignal = LINSignalCode * Scale + Offset, \quad if \quad LINSignalCode \in [MinValue;\ MaxValue]. \qquad (1)$$

If the output signal is not updated within the *Output Signal Autoreset Time*, the output signal will be reset to undefined. The auto-reset is disabled when the *Output Signal Autoreset Time* configuration parameter is equal to 0.

The initial value of the *LIN Signal* function block output signal is undefined.

1.3.2.2. Transmitting LIN Signals

When the LIN Signal function block transmits signals on the LIN bus, the signal data are acquired from the function block signal input source and processed, before transmission, in a similar way as the incoming LIN signals:

- Logical signals are transmitted only when they are in the [*MinValue; MaxValue*] range;
- BCD signals are transmitted without any conversion;
- ASCII signals are masked with 0xFF value and then transmitted;
- The physical signals are converted to the LIN signal code using Scale and Offset configuration parameters and then saturated to the *MinValue* or *MaxValue* if the code goes out of the [*MinValue; MaxValue*] range.
- Undefined signals are presented by their initial signal values.

1.3.3. LIN Slave Response

There is a special *LIN Slave Response* function block with one output signal.

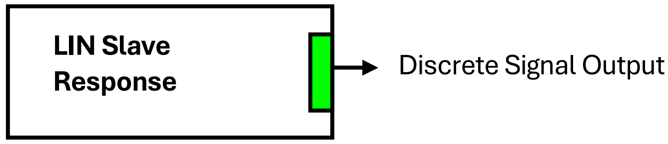


Figure 4 – LIN Slave Response Function Block

This function block does not have any configuration parameters. It provides a discrete logical output signal reflecting the state of the LIN slave node according to the LIN standard requirements.

In the master mode, this function block is not used, and its output signal is undefined.

1.3.4. LIN Unconditional Frame

There are *25 LIN Unconditional Frame* function blocks available to the user. Each function block represents one LIN frame that can be transmitted or received on the LIN bus.

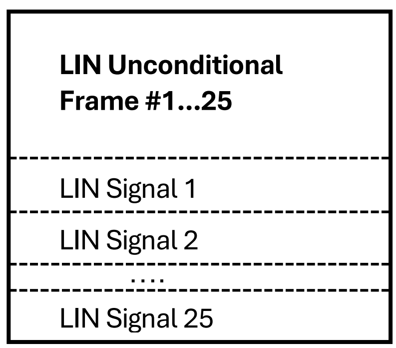


Figure 5 – LIN Unconditional Frame Function Block

There are no signal inputs or outputs in this function block; all data is transmitted or received through the associated *LIN Signal* function blocks. There can be up to 25 LIN signals associated with each LIN frame.

Configuration parameters of the *LIN Unconditional Frame* function block are presented below:

Table 6 – LIN Unconditional Frame Configuration Parameters

Name	Default Value[1]	Range	Units	Description
LIN Frame Kind	Publish	{Undefined, Publish, Subscribe}	–	Defines whether the frame is transmitted or received.
Frame ID	0x12	0...0x3F	–	Frame ID.
Associated with Event Triggered Frame	No	{No, Yes}	–	Defines whether this Frame is used in the Event Triggered Frame.
Size	4	[1...8]	Byte	Frame Size.
Checksum Type	Enhanced	{Classic, Enhanced}	–	Type of the frame checksum.
Signal #1 Number	1	[0...35]	–	Number of the 1-st *LIN Signal* function block.
Signal #1 Offset	0	[0...63]	Bit	Offset of the LIN signal defined by the 1-st *LIN Signal* function block.
Signal #2 Number	2	[0...35]	–	Number of the 2-nd *LIN Signal* function block.
Signal #2 Offset	8	[0...63]	Bit	Offset of the LIN signal defined by the 2-nd *LIN Signal* function block.
...	...	...	...	...
Signal #25 Number	0	[0...35]	–	Number of the 15-th *LIN Signal* function block.
Signal #25 Offset	0	[0...63]	Bit	Offset of the LIN signal defined by the 15-th *LIN Signal* function block.

The *LIN Conditional* function blocks are numbered from 1 to 15. When the *Signal #1…25 Number* is equal to 0, the *LIN Conditional* function block is not defined and the associated *Signal #1…25 Offset* configuration parameter is not used.

1.3.5. LIN Event Triggered Frame

There is one *LIN Event Triggered Frame* function block available to the user. It can contain up to 5 associated LIN frames. There are no signal inputs or outputs in this function block.

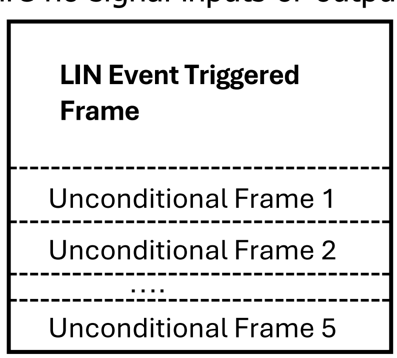


Figure 6 – LIN Event Triggered Frame Function Block

Configuration parameters of the *LIN Event Triggered Frame* function block are presented below:

Table 7 – LIN Event Triggered Frame Configuration Parameters

Name	**Default Value**	**Range**	**Units**	**Description**
LIN Frame Kind	Undefined	{Undefined, Publish, Subscribe}	–	Defines whether the frame is transmitted or received.
Frame ID	0	0...0x3F	–	Frame ID.
Size	1	[1...8]	Byte	Frame Size.
Checksum Type	Classic	{Classic, Enhanced}	–	Type of the frame checksum.
Collision Resolving Schedule Table Number	0	[0...1]	–	Number of the *Collision Resolving Schedule Table* function block used in case of a collision.
Unconditional Frame #1 Number	0	[0...10]	–	Number of the 1-st *LIN Unconditional Frame* function block.
Unconditional Frame #2 Number	0	[0...10]	–	Number of the 2-nd *LIN Unconditional Frame* function block.
...	...	...	...	...
Unconditional Frame #5 Number	0	[0...10]	–	Number of the 5-th *LIN Unconditional Frame* function block.

Collision Resolving Schedule Table function blocks are numbered starting from 0. When the *Collision Resolving Schedule Table Number* is equal to 0, the function block is not defined. The same rule applies to the *LIN Unconditional Frame* function blocks.

When the event triggered frame is defined, all associated unconditional frames should have the same frame size and checksum type as defined in the *LIN Event Triggered Frame* function block. They should also have *Associated with Event Triggered Frame* configuration parameter set to Yes.

1.3.6. LIN Sporadic Frame

There is one *LIN Sporadic Frame* function block available to the user. It can contain up to 5 associated LIN frames. There are no signal inputs or outputs in this function block.

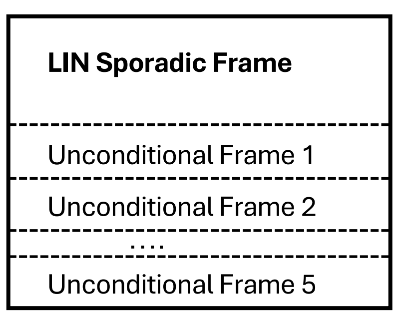


Figure 7 – LIN Sporadic Frame Function Block

This block is used only by master nodes. See: *Node Type* configuration parameter in the *LIN Common* function block.

Configuration parameters of the *LIN Sporadic Frame* function block are presented below:

Table 8 – LIN Sporadic Frame Configuration Parameters

Name	Default Value	Range	Units	Description
LIN Frame Kind	Publish	{Undefined, Publish, Subscribe}	–	Defines whether the frame is transmitted or received.
Unconditional Frame #1 Number	1	[0...10]	–	Number of the 1-st *LIN Unconditional Frame* function block.
Unconditional Frame #2 Number	0	[0...10]	–	Number of the 2-nd *LIN Unconditional Frame* function block.
...	...	...	...	...
Unconditional Frame #5 Number	0	[0...10]	–	Number of the 5-th *LIN Unconditional Frame* function block.

Sending priorities in the sporadic frame is defined in descending order: unconditional frame #1 has the maximum priority and unconditional frame #5 – minimum. When *Unconditional Frame #1…5 Number* is equal to 0, the frame is undefined.

1.3.7. LIN Main Schedule Table

There is one *Main Schedule Table* function block available to the user. It is used by the master node by default and can contain up to 10 schedule entries. There are no signal inputs and outputs in this function block.

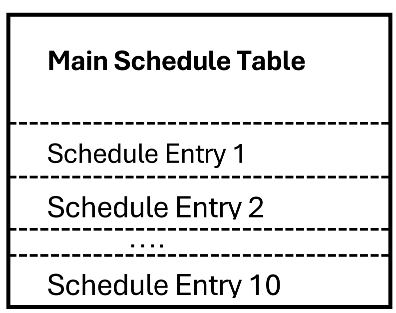


Figure 8 – LIN Main Schedule Table Function Block

Configuration parameters of the *Main Schedule Table* function block are presented below:

Table 9 – LIN Main Schedule Table Configuration Parameters

Name	Default Value	Range	Units	Description
Entry #1 Frame Type	Unconditional	{Undefined, Unconditional, Event Triggered, Sporadic}	–	1-st schedule entry frame type.
Entry #1 Frame Number	2	Depends on the frame type	–	1-st schedule entry frame number.
Entry #1 Delay	20	[0…10000]	ms	1-st schedule entry delay.
Entry #2 Frame Type	Unconditional	{Undefined, Unconditional, Event Triggered, Sporadic}	–	2-nd schedule entry frame type.
Entry #2 Frame Number	3	Depends on the frame type	–	2-nd schedule entry frame number.
Entry #2 Delay	20	[0…10000]	ms	2-nd schedule entry delay.
…	…	…	…	…
Entry #10 Frame Type	Undefined	{Undefined, Unconditional, Event Triggered, Sporadic}	–	10-th schedule entry frame type.
Entry #10 Frame Number	0	Depends on the frame type	–	10-th schedule entry frame number.
Entry #10 Delay	0	[0…10000]	ms	10-th schedule entry delay.

When the frame type is undefined, the schedule entry is skipped. When the frame number is equal to 0, the schedule entry is empty. When the schedule table is used, the first schedule entry should be defined.

1.3.8. LIN Collision Schedule Table

There is one *Collision Schedule Table* function block available to the user. It is used by the master node to resolve collisions in the event triggered frames. *Collision Schedule Table* function block has the same structure as the *Main Schedule Table*. The only difference is in the number of the schedule table entries.

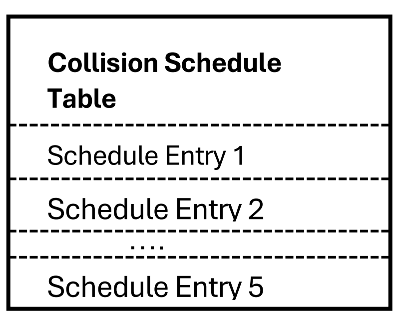


Figure 9 – LIN Collision Schedule Table Function Block

The collision schedule table can contain up to 5 schedule entries. The schedule entries follow the same rules defined for the *Main Schedule Table* with the exception that only unconditional frames are allowed in the collision schedule table.

Configuration parameters of the *Collision Schedule Table* function block are presented below:

Table 10 – LIN Collision Table Configuration Parameters

Name	Default Value	Range	Units	Description
Entry #1 Frame Type	Undefined	{Undefined, Unconditional}	–	1-st schedule entry frame type.
Entry #1 Frame Number	0	[0...10]	–	1-st schedule entry frame number.
Entry #1 Delay	0	[0...10000]	ms	1-st schedule entry delay.
Entry #2 Frame Type	Undefined	{Undefined, Unconditional}	–	2-nd schedule entry frame type.
Entry #2 Frame Number	0	[0...10]	–	2-nd schedule entry frame number.
Entry #2 Delay	0	[0...10000]	ms	2-nd schedule entry delay.
...	...	...	...	...
Entry #5 Frame Type	Undefined	{Undefined, Unconditional}	–	5-th schedule entry frame type.
Entry #5 Frame Number	0	[0...10]	–	5-th schedule entry frame number.
Entry #5 Delay	0	[0...10000]	ms	5-th schedule entry delay.

1.4. Binary Functions

There are fifteen *Binary Function* blocks available to the user for performing simple data conversions. Each *Binary Function* block has two continuous signal inputs and one continuous signal output.

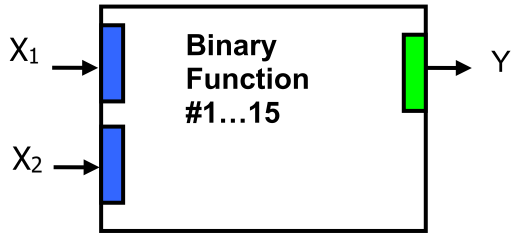


Figure 10 – Binary Function Block

The *Binary Function* block performs the following data conversion:

$$Y = A \cdot F[a_1 \cdot f_1(X_1) + b_1; a_2 \cdot f_2(X_2) + b_2] + B, \; n = 1{,}2; \qquad (2)$$

where:
- X_n – Input signal;
- $f_n(X_n)$ – Unary function;
- a_n – Scale;
- b_n – Offset;
- $F[x;y]$ – Binary Function;
- A – Output Scale;
- B – Output Offset.

The function block input signals can be undefined. The user can specify a default signal value that will be used when the signal is not defined. If the default signal value is not specified, the output signal of the function block will also become undefined.

The following unary functions can be used to process the input signals.

Table 11 – Unary Functions

Value	Function Name	Description	Comment
0	Undefined	f(x) = x	Signal is not processed
1	! Logical Not	f(x) = !x	x is converted into 4-byte unsigned integer before function is applied
2	~ Bitwise Not	f(x) = ~x	x is converted into 4-byte unsigned integer before function is applied
3	abs(x) Absolute	f(x) = x, if x≥0 f(x) = -x, if x<0	

The following binary functions are defined in the function block:

Table 12 – Output Binary Functions

Value	**Function Name**	**Description**	**Comment**
0	Undefined	F[x;y] = Undefined	Output signal is undefined
1	+ Addition	F[x;y] = x + y	
2	- Subtraction	F[x;y] = x - y	
3	* Multiplication	F[x;y] = x * y	
4	/ Division	F[x;y] = x / y	Division by 0 gives 0
5	% Modulus	F[x;y] = x % y	x and y are converted into 4-byte unsigned integers before function is applied
6	max(x,y) Maximum	F[x;y] = x, if x≥y F[x;y] = y, if x<y	
7	min(x,y) Minimum	F[x;y] = x, if x≤y F[x;y] = y, if x>y	
8	== Equal	F[x;y] = 1, if x=y F[x;y] = 0, if x≠y	
9	!= Not Equal	F[x;y] = 1, if x≠y F[x;y] = 0, if x=y	
10	> Great	F[x;y] = 1, if x>y F[x;y] = 0, if x≤y	
11	>= Great Equal	F[x;y] = 1, if x≥y F[x;y] = 0, if x<y	
12	< Less	F[x;y] = 1, if x<y F[x;y] = 0, if x≥y	
13	<= Less Equal	F[x;y] = 1, if x≤y F[x;y] = 0, if x>y	
14	|| Logical OR	F[x;y] = x ∨ y	x and y are converted into 4-byte unsigned integers before function is applied
15	&& Logical AND	F[x;y] = x ∧ y	x and y are converted into 4-byte unsigned integers before function is applied
16	| Bitwise OR	F[x;y] = x | y	x and y are converted into 4-byte unsigned integers before function is applied
17	& Bitwise AND	F[x;y] = x & y	x and y are converted into 4-byte unsigned integers before function is applied
18	^ Bitwise XOR	F[x;y] = x ^ y	x and y are converted into 4-byte unsigned integers before function is applied
19	<< Left Shift	F[x;y] = x << y	x and y are converted into 4-byte unsigned integers before function is applied
20	>> Right Shift	F[x;y] = x >> y	x and y are converted into 4-byte unsigned integers before function is applied

The *Binary Function* block has the following set of configuration parameters:

Table 13 – Binary Function Configuration Parameters

Name	Default Value	Range	Units	Description
Binary Function	Undefined	See Table 12	–	F[x;y] – Binary function
Output Scale	1	Any value	–	A – Output Scale
Output Offset	0	Any value	–	B – Output Offset
Input #1 Signal Source	Not Connected	Any signal output of any function block or “Not Connected”	–	X_1 – Input Signal #1
Input #1 Signal Number	0	Depends on 30x0h source	–	X_1 – Input Signal #1
Input #1 Signal Default	No	{No, Yes}	–	Defines whether the default signal value for X_1 is defined.
Input #1 Signal Default Value	0	Any value	–	X_1 default value, if *Input #1 Signal Default* is *Yes*.
Unary Function #1	Undefined	See Unary Function table	–	$f_1(x)$ – Unary function #1
Scale #1	1	Any value	–	a_1 – Scale #1
Offset #1	0	Any value	–	b_1 – Offset #1
Input #2 Signal Source	Not Connected	Any signal output of any function block or “Not Connected”	–	X_2 – Input Signal #2
Input #2 Signal Number	0	Depends on 30x0h source	–	X_2 – Input Signal #2
Input #2 Signal Default	No	{No, Yes}	–	Defines whether the default signal value for X_2 is defined.
Input #2 Signal Default Value	0	Any value	–	X_2 default value, if *Input #2 Signal Default* is *Yes*.
Unary Function #2	Undefined	See Unary Function table	–	$f_2(x)$ – Unary function #2
Scale #2	1	Any value	–	a_2 – Scale #2
Offset #2	0	Any value	–	b_2 – Offset #2

1.5. DTC React

DTC React is a function block that allows the ECU to receive and process the DM1 messages. There are 16 separated function blocks that can capture up to 16 different DM1 messages. Each DTC React has two mandatory and 2 optional parameters. The mandatory parameters are the SPN and FMI. If only these parameters are used, the output will be set to high if the DM1 message with the combination of selected SPN and FMI. The state will remain high for five seconds and will be set if the DM1 message will be received again.

Among optional parameters there are lamp setting and the source address. To enable them, the "**Lamp Used to Trigger Reaction**" and "**Source Address Used to Trigger Reaction**" should be set to 1, *True*. In this case, beside SPN and FMI the ECU will compare the Lamp Setting and/or Source Address of the received message.

The exceptions are the following SPN:

- SPN 1213 and Lamp Status 0x40;
- SPN623 and Lamp Status 0x10;
- SPN624 and Lamp Status 0x04;
- SPN987 and Lamp Status 0x01;

In case if the SPNs above are chosen, the DTC React function block will set the output to HIGH if SPN and Lamp Status matches even if FMI doesn't match. However, if the "**Source Address Used to Trigger Reaction**" is set to 1, *True* and selected address doesn't match, the DTC React output will be set to FALSE.

1.6. Diagnostic Function Blocks

The ECU 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 which is a combination of:

SPN	Suspect Parameter Number	(user defined)
FMI	Failure Mode Identifier	(see Table 23)
CM	Conversion Method	(always set to 0)
OC	Occurrence Count	(number of times the fault has happened)

In addition to supporting the DM1 message, ECU 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

Fault detection and reaction is a standalone functionality that can be configured to monitor and report diagnostics of various controller parameters. The Controller supports 16 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.

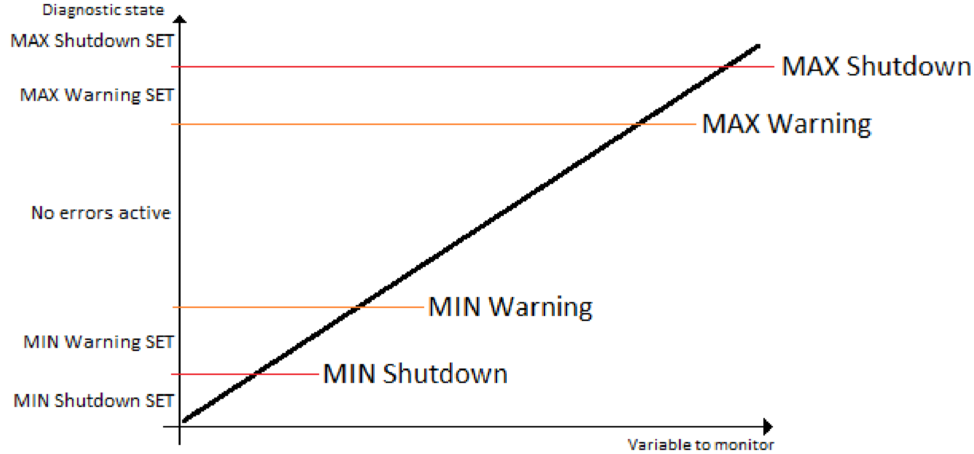


Figure 11 – Double Minimum and Maximum Error Thresholds

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 Outputs**” 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 ECU 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).

At power up, the DM1 message will not be broadcasted until after 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 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 it 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 21. By default, the '*Amber, Warning'* lamp is typically the one set be any active fault.

Table 14 – Lamp Set by Event in DM1 Options

0	*Protect*
1	*Amber Warning*
2	*Red Stop*
3	*Malfunction*

"**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.

Table 15 – FMI for Event Options

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

Every fault has associated a default FMI with them. The used FMI can be configured with "**FMI for Event**" setpoint, presented in Table 15. When an FMI is selected from Low Fault FMIs in Table 23 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 23. There is no automatic setting of High and Low FMIs in the firmware, the user can configure these freely.

Table 16 – Low Fault FMIs and corresponding High Fault FMIs

Low Fault FMIs	High Fault FMIs
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*

1.7. CAN Output Function Block

The CAN Transmit function block is used to send any output from another function block (i.e. input, CAN receive) to the J1939 network. The AX140620 ECU has 25 CAN Output Messages and each message has four completely user defined signals.

1.7.1. CAN Output Message Setpoints

Each CAN Transmit Message setpoint group includes setpoints that effect the whole message and are thus mutual for all signals of the message. These setpoints are presented in this section. The setpoints that configure an individual signal are presented in next section.

"**CAN Interface**" setpoint is used to define which of the two CAN Interfaces is used to transmit the message in question.

The "**Transmit PGN**" setpoint sets PGN used with the message. **User should be familiar with the SAE J1939 standard, and select values for PGN/SPN combinations as appropriate from section J1939/71**.

"**Repetition Rate**" setpoint defines the interval used to send the message to the J1939 network. If the "**Repetition Rate**" is set to zero, the message is disabled unless it shares its PGN with another message. In case of a shared PGN repetition rate of the LOWEST numbered message are used to send the message 'bundle'.

> 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.

By default, all messages are sent on Proprietary B PGNs as broadcast messages. Thus "**Transmit Message Priority**" is always initialized to 6 (low priority) and the "**Destination Address**" setpoint is not used. This setpoint is only valid when a PDU1 PGN has been selected, and it can be set either to the Global Address (0xFF) for broadcasts, or sent to a specific address as setup by the user.

"**Transmit on LIN Unconditional Frame Number"** setpoint allows ECU to send the CAN Output message upon reception of a specific LIN Unconditional Frame. When this parameter is set to a non-zero value, the CAN frame will be sent and the transmit timer will be reset. If "**Repetition Rate**" is set to 0, the ECU will still send the CAN frame.

1.7.2. CAN Output Signal Setpoints

Each CAN transmit message has four associated signals, which define data inside the Transmit message. “**Control Source**” setpoint together with “**Control Number**” setpoint define the signal source of the message. “**Control Source**” and “**Control Number**” options are listed in Table 1. Setting “**Control Source**” to ‘*Control Not Used’* disables the signal.

“**Transmit Data Size**” setpoint determines how many bits signal reserves from the message. “**Transmit Data Index in Array**” determines in which of 8 bytes of the CAN message LSB of the signal is located. Similarly “**Transmit Bit Index in Byte**” determines in which of 8 bits of a byte the LSB is located. These setpoints are freely configurable, thus **it is the user’s responsibility to ensure that signals do not overlap and mask each other**.

“**Transmit Data Resolution**” setpoint determines the scaling done on the signal data before it is sent to the bus. “**Transmit data Offset**” setpoint determines the value that is subtracted from the signal data before it is scaled. Offset and Resolution are interpreted in units of the selected source signal.

1.8. CAN Input 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 (i.e. Outputs).

"**CAN Interface**" setpoint is used to define from which of the two CAN Interfaces the message in question is received.

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 off the bud within the "**Receive Message Timeout**" period. This could trigger a Lost Communication event as described in section 1.9. 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 trigger a Lost Communication fault.

By default, all control messages are expected to be sent to the ECU on Proprietary B PGNs. However, should a PDU1 message be selected, the ECU 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 clock can be selected as the source of the control input for the output function blocks. When this is 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 thesholds 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 Controller supports up to 25 unique CAN Receive Messages. Defaults setpoint values are listed in section 4.12.

1.9. CAN Common

The CAN Common function block contains additional settings for the CAN bus. It allows the ECU to enable/disable the Auto Baud Rate feature, set a needed baud rate, and to control the DM1 message broadcast. For more details, see the following table.

Table 17 – CAN Common Function Block Parameters.

Name	Default Value	Range	Units	Description
Baud Rate	Current Baud Rate	[125, 250, 500, 667, 1000]	kbit/s	CAN baud rate. If the "Auto Baud Rate" is set to 0, False, this setpoint will define the default baud rate the ECU will stick to.
Auto Baud Rate	1, True	[0...1]		Defines if ECU will perform the auto baud rate scan or it will connect the *"Baud Rate"* setpoint value.
CAN Send empty DM1 messages	0, Diagnostic Messages Off	[0...2]		Defines DM1 messages behavior. Can be 0, Diagnostic Messages Off; 1, Disable Empty DM1 Messages; 2, Diagnostic Messages On.

1.10. CAN Interface

The CAN interface is compliant with Bosch CAN protocol specification, Rev.2.0, Part B, and the following J1939 standards:

Table 18. CAN Standard Implementation

ISO/OSI Network Model Layer	J1939 Standard
Physical	J1939/11 – Physical Layer, 250K bit/s, Twisted Shielded Pair. Rev. SEP 2006. J1939/15 - Reduced Physical Layer, 250K bits/sec, Un-Shielded Twisted Pair (UTP). Rev. AUG 2008. J1939/14 - Physical Layer, 500 Kbps. Rev. OCT 2011. J1939/16 – Automatic Baud Rate Detection Process. Rev. NOV 2018.
Data Link	J1939/21 – Data Link Layer. Rev. DEC 2006
	The converter supports Transport Protocol for Commanded Address messages (PGN 65240), ECU identification messages -ECUID (PGN 64965), and software identification messages -SOFT (PGN 65242). It also supports responses on PGN Requests (PGN 59904). Please note that the Proprietary A PGN (PGN 61184) is taken by Axiomatic Simple Proprietary Protocol and is not available for the user.
Network	J1939, Appendix B – Address and Identity Assignments. Rev. FEB 2010. J1939/81 – Network Management. Rev. MAR2017.
	The converter is an Arbitrary Address Capable ECU. It can dynamically change its network address in real time to resolve an address conflict with other ECUs. The converter supports: Address Claimed Messages (PGN 60928), Requests for Address Claimed Messages (PGN 59904) and Commanded Address Messages (PGN 65240).
Transport	N/A in J1939.
Session	N/A in J1939.
Presentation	N/A in J1939.
Application	J1939/71 – Vehicle Application Layer. Rev. APR 2014 with J1939DA – Digital Annex. Rev. OCT 2014.
	The converter can receive and transmit application specific PGNs. All application specific PGNs are user programmable.
	J1939/73 – Application Layer – Diagnostics. Rev. FEB 2010
	Memory access protocol (MAP) supports: DM14, DM15, DM16 messages used by the Axiomatic EA to program configuration parameters.

1.10.1. CAN Baud Rate

The converter can operate at J1939 standard 250 kbit/s and 500 kbit/s baud rates. It can also run at 125kbit/s, 667 kbit/s and 1Mbit/s – the maximum baud rate supported by the CAN hardware.

The baud rate selection is performed automatically upon connection to the CAN network using passive and active automatic baud rate detection process described in J1939/16. Once detected, the baud rate is stored in non-volatile memory and used on the next converter power-up.

The baud rate detection can be disabled for permanently installed units to maintain the desired baud rate on the CAN network.

1.10.2. J1939 Name and Address

Before sending and receiving any application data, the converter claims its network address with a unique J1939 Name. The Name fields are presented in the table below:

Table 19. J1939 Name Fields

Field Name	Field Length	Field Value	Configurable
Arbitrary Address Capable	1 bit	1 (Capable)	No
Industry Group	3 bit	0 (Global)	No
Vehicle System Instance	4 bit	0 (First Instance)	No
Vehicle System	7 bit	0 (Nonspecific System)	No
Reserved	1 bit	0	No
Function	8 bit	25 (Network Interconnect ECU)	No
Function Instance	5 bit	25 (AX140620, LIN – J1939 CAN, Axiomatic with PWM Output proprietary)	No
ECU Instance	3 bit	0 (First Instance)	Yes
Manufacturer Code	11 bit	162 (Axiomatic Technologies Corp.)	No
Identity Number	21 bit	Calculated on the base of the microcontroller unique ID	No

The user can change the converter *ECU Instance* using the Axiomatic EA to accommodate multiple converters on the same CAN network.

The converter takes its network *ECU Address* from a pool of addresses assigned to self-configurable ECUs. The default address can be changed during an arbitration process or upon receiving a commanded address message. The new address value will be stored in a non-volatile memory and used next time for claiming the network address. The *ECU Address* can also be changed in the Axiomatic EA.

1.10.3. Network Bus Terminating Resistors

The converter does not have an embedded 120 Ohm CAN bus terminating resistor.

Terminating resistors should be installed externally on both ends of the CAN twisted pair cable according to the J1939/11(15) standards to avoid communication errors.

Even if the length of the CAN network is short and the signal reflection from both ends of the cable can be ignored, at least one 120 Ohm resistor is required for the majority of CAN transceivers to operate properly.

2. FLASHING NEW FIRMWARE

When the new firmware becomes available, the user can replace the converter firmware in the field using the unit embedded bootloader. The firmware file can be received from Axiomatic on request.

To flash the new firmware, the user should activate the embedded bootloader. To do so, start the Axiomatic EA and in the *Bootloader Information* group screen click on the *Force Bootloader to Load on Reset* parameter. The following dialog will appear, see Figure 45.

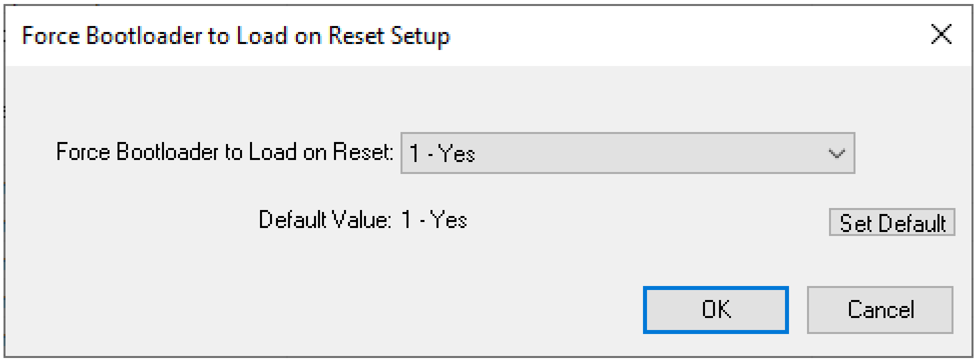


Figure 12. Bootloader Activation. First Step

The EA will prompt the user to change the *Force Bootloader to Load on Reset* parameter flag to *Yes*. This will automatically activate the bootloader on the next ECU reset. After accepting the change, the next screen will ask the user if the reset is actually required, see Figure 46. Select *Yes*.

After automatic reset, instead of *AX140620, LIN - CAN Protocol Converter*, the user will see *J1939 Bootloader* ECU in the *J1939 CAN Network* top-level group in the EA. This means that the bootloader is activated and ready to accept the new firmware.

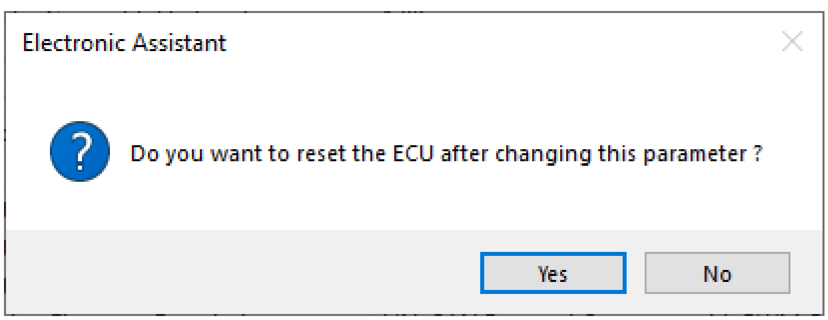


Figure 13. Bootloader Activation. Final Reset

All the bootloader specific information: converter hardware, bootloader details, and the currently installed application firmware remains the same in the bootloader mode and the user can read it in the *Bootloader Information* group screen, see Figure 47. The information can be slightly different for different versions of the bootloader.

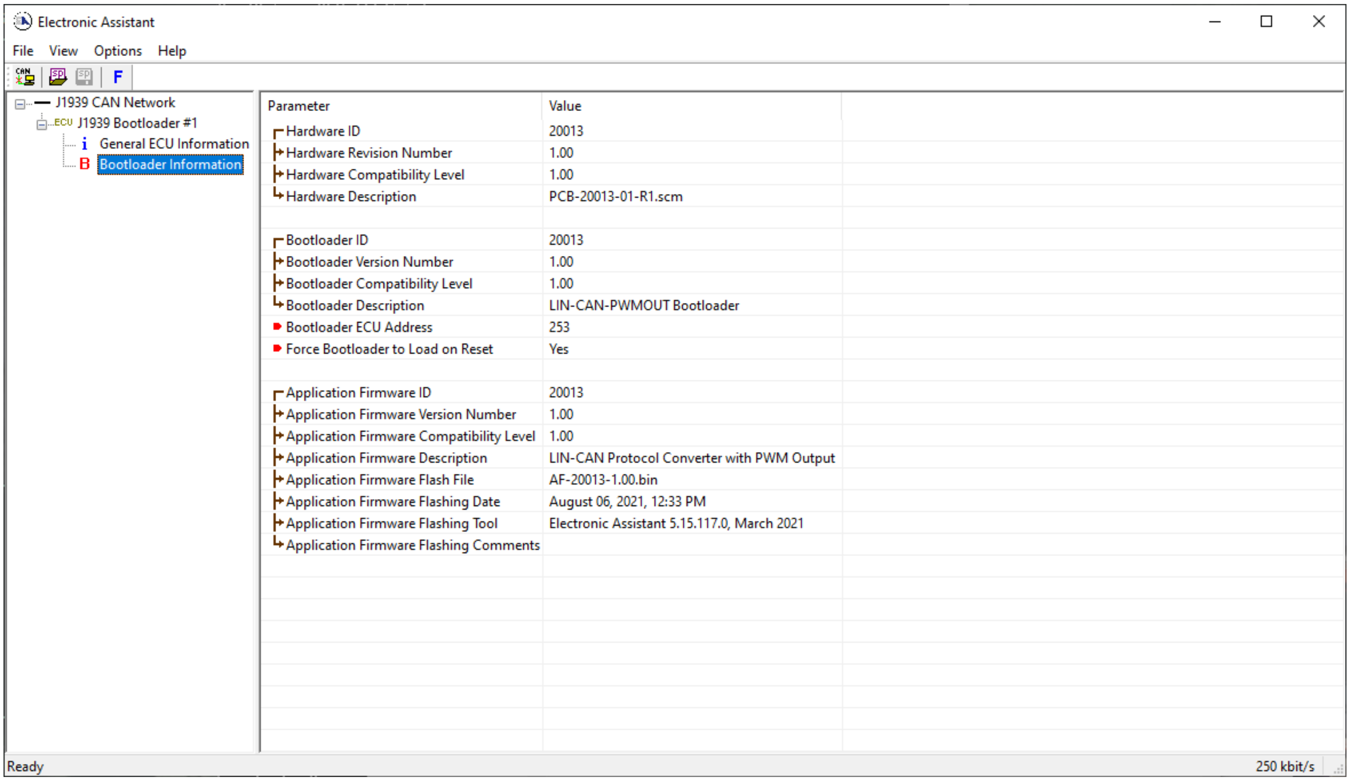


Figure 14. Bootloader Information Screen

At this point, the user can return to the installed converter firmware by changing the *Force Bootloader to Load on Reset* flag back to *No* and resetting the ECU.

To flash the new firmware, the user should click on F toolbar icon or from the *File* menu select the *Open Flash File* command. The *Open Application Firmware Flash File* dialog will appear. Pick up the flash file with the new converter firmware and confirm the selection by pressing the *Open* button. The *Flash Application Firmware* dialog window will appear[1], see Figure 48.

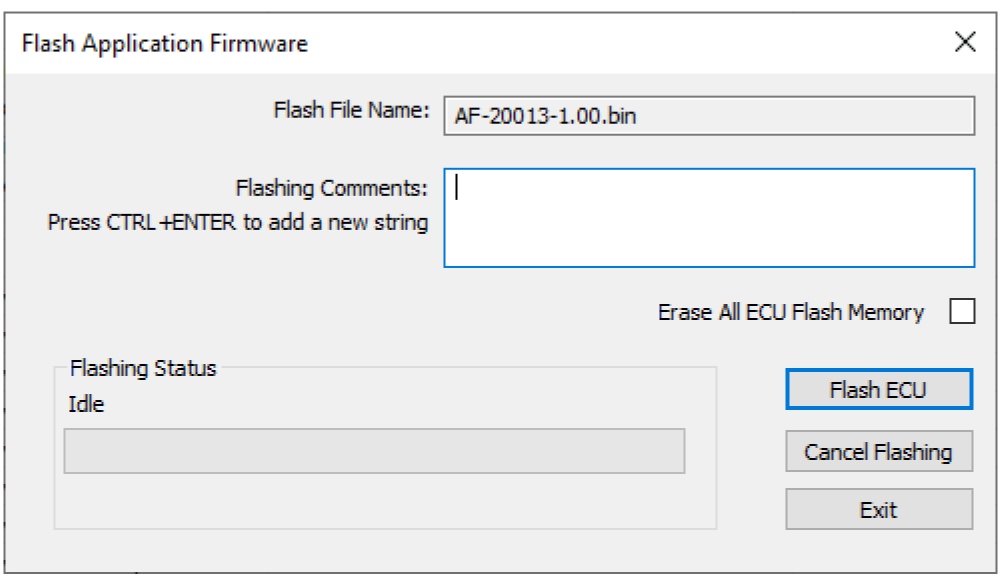


Figure 15. Flashing New Firmware. Preparation

[1] *In this example, instead of the new firmware, the old firmware V1.00 is being simply re-flashed.*

Now the user can add any comments to the flashing operation in the *Flashing Comments* field. They will be stored in the *Bootloader Information* group after flashing.

The user can also check the *Erase All ECU Flash Memory* flag to erase all configuration parameters related to LIN set by the old firmware and force the converter to load the LIN default values after flashing the new firmware. This operation, used in other products to reset all configuration parameters kept in the flash memory to their default values, has no effect in this product. This is because all the configuration parameters of the converter other than LIN are stored in a separate EEPROM memory.

Otherwise, the default LIN values will be set only to the new configuration parameters introduced in the new firmware. The old configuration parameters will keep their original values unless otherwise is stated in the user manual.

Select the *Flash ECU* button to start flashing. A reminder that the old application firmware will be destroyed by the flashing operation will appear. Press *Ok* to continue and watch the dynamics of the flashing operation in the *Flashing Status* field. When flashing is done, the following screen will appear prompting the user to reset the ECU, see Figure 49.

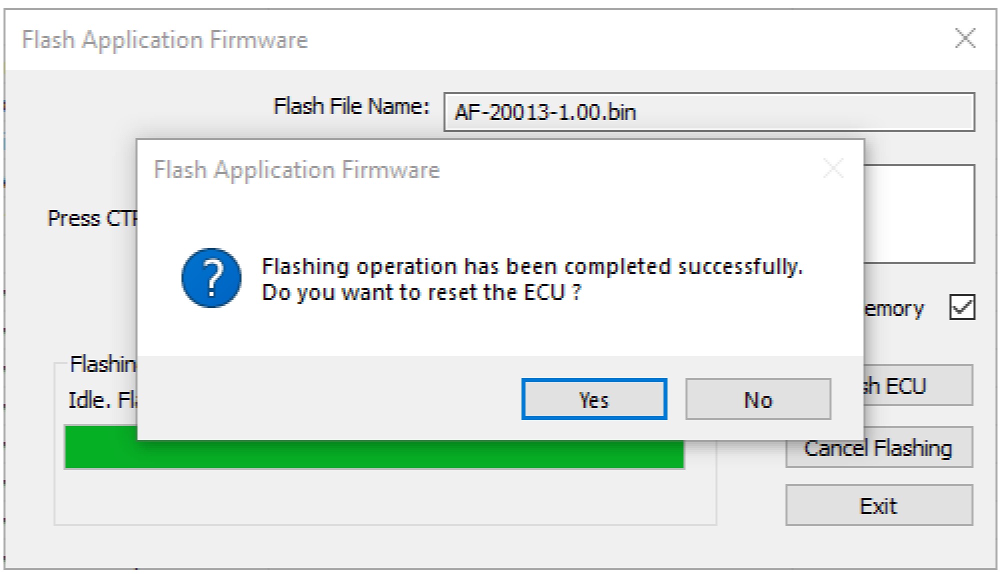


Figure 16. Flashing New Firmware. Final Reset.

Select *Yes* and see the ECU running the new firmware, see Figure 50. This will indicate that the flashing operation has been performed successfully.

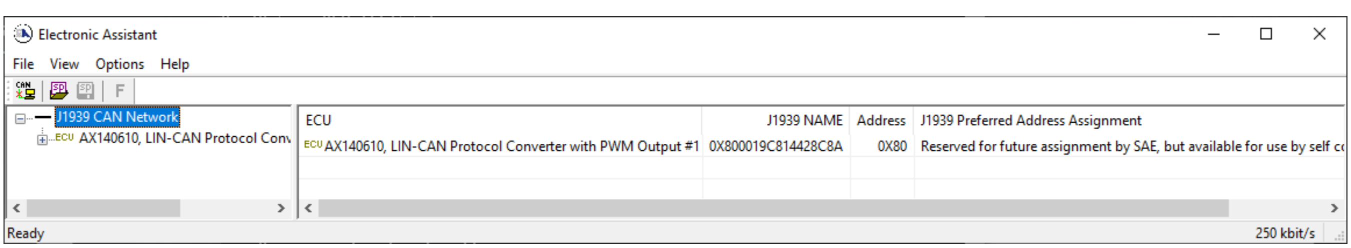


Figure 17. Firmware has been Updated. New Firmware Screen

For more information, see the *J1939 Bootloader* section of the Axiomatic EA user manual.

3. OVERVIEW OF J1939 FEATURES

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

- Configurable ECU Instance in the NAME (to allow multiple ECUs on the same network)
- Configurable Input Parameters
- Configurable PGN and Data Parameters
- Configurable Diagnostic Messaging Parameters, as required
- Diagnostic Log, maintained in non-volatile memory

3.1. Introduction to Supported Messages

The ECU is compliant with the standard SAE J1939, and supports following PGNs from the standard.

From J1939-21 – Data Link Layer

• Request		59904	0x00EA00
• Acknowledgement		59392	0x00E800
• Transport Protocol – Connection Management		60416	0x00EC00
• Transport Protocol – Data Transfer Message		60160	0x00EB00
• Proprietary B	from	65280	0x00FF00
	to	65535	0x00FFFF

From J1939-73 – Diagnostics

• DM1 – Active Diagnostic Trouble Codes	65226	0x00FECA
• DM2 – Previously Active Diagnostic Trouble Codes	65227	0x00FECB
• DM3 – Diagnostic Data Clear/Reset for Previously Active DTCs	65228	0x00FECC
• DM11 – Diagnostic Data Clear/Reset for Active DTCs	65235	0x00FED3

From J1939-81 – Network Management

• Address Claimed/Cannot Claim	60928	0x00EE00
• Commanded Address	65240	0x00FED8

From J1939-71 – Vehicle Application Layer

• Software Identification	65242	0x00FEDA
• Software Identification	65242	0x00FEDA
• Component Identification	65259	0x00FEEB

None of the application layer PGNs are supported as part of the default configurations, but they can be selected as desired for transmit 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 CAN network.

3.2. NAME, Address and Software ID

The ECU has the following default 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	127, I/O Controller
Function Instance	7, Axiomatic AX031220
ECU Instance	0, First Instance
Manufacture Code	162, Axiomatic Technologies
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 from one another when they are connected on the same network.

The default value of the “ECU Address” setpoint is 128 (0x80), which is the preferred starting address for self-configurable ECUs as set by the SAE in J1939 tables B3 and B7. The Axiomatic EA supports the selection of any address between 0 and 253. ***It is 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 ECU will continue select the next highest address until it finds one that it can claim. See J1939/81 for more details about address claiming.

ECU Identification Information

PGN 64965 ECU Identification Information -ECUID

Transmission Repetition Rate: On request

Data Length:	Variable
Extended Data Page:	0
Data Page:	0
PDU Format:	253
PDU Specific:	197 PGN Supporting Information:
Default Priority:	6
Parameter Group Number:	64965 (0x00FDC5)

Start Position	Length	Parameter Name	SPN
a	Variable	ECU Part Number, Delimiter (ASCII “*”)	2901
b	Variable	ECU Serial Number, Delimiter (ASCII “*”)	2902
c	Variable	ECU Location, Delimiter (ASCII “*”)	2903
d	Variable	ECU Type, Delimiter (ASCII “*”)	2904
e	Variable	ECU Manufacturer Name, Delimiter (ASCII “*”)	4304

(a)*(b)*(c)*(d)*(e)*

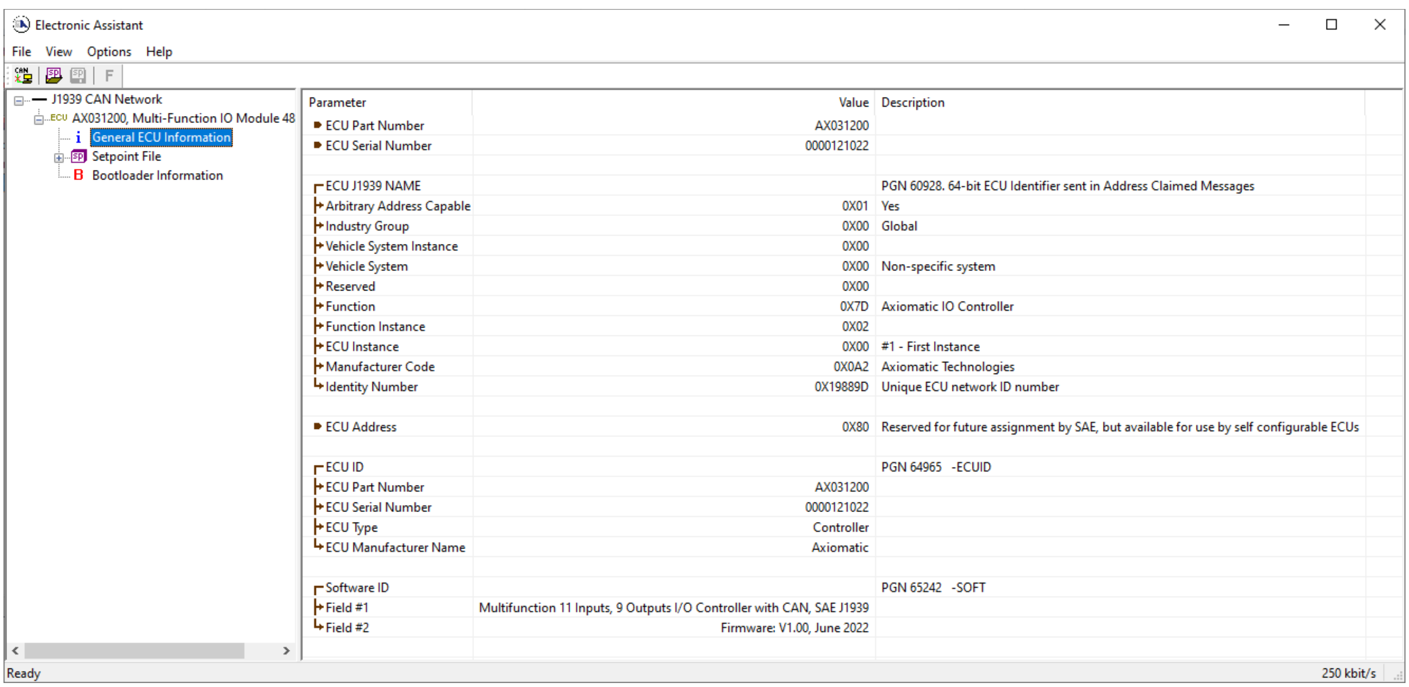


Figure 18 – General ECU Information

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 21 In 6 Out CAN Controller, 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.

Component Identification

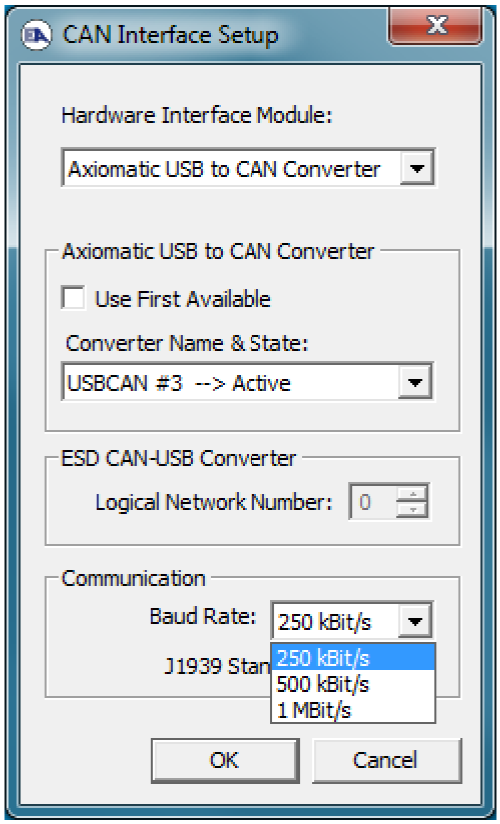
PGN 65259		Component Identification	-CI
Transmission Repetition Rate:		On request	
Data Length:		Variable	
Extended Data Page:		0	
Data Page:		0	
PDU Format:		254	
PDU Specific:		235 PGN Supporting Information:	
Default Priority:		6	
Parameter Group Number:		65259 (0x00FEEB)	
Start Position	Length	Parameter Name	SPN
a	1-5 Byte	Make, Delimiter (ASCII "*")	586
b	Variable	Model, Delimiter (ASCII "*")	587
c	Variable	Serial Number, Delimiter (ASCII "*")	588
d	Variable	Unit Number (Power Unit), Delimiter (ASCII "*")	233
(a)*(b)*(c)*(d)*(e)*			

4. ECU SETPOINTS ACCESSED WITH THE AXIOMATIC ELECTRONIC ASSISTANT

This section describes in detail each setpoint, and their default and ranges. The setpoints are divided into setpoint groups as they are shown in the Axiomatic EA. For more information on how each setpoint is used by 10 Analog Input, refer to the relevant section in this user manual.

4.1. Accessing the ECU Using the Axiomatic Electronic Assistant

ECU does not need any specific setup for the Axiomatic EA. The CAN Interface Setup can be found from “Options” menu in the Axiomatic EA. Please refer UMAX07050x **Connecting to the J1939 Bus** section for Axiomatic Electronic Assistant CAN Interface Setup instructions.



4.2. J1939 Network Parameters

"ECU Instance Number" and "ECU Address" setpoints and their effect are defined in section 3.2.

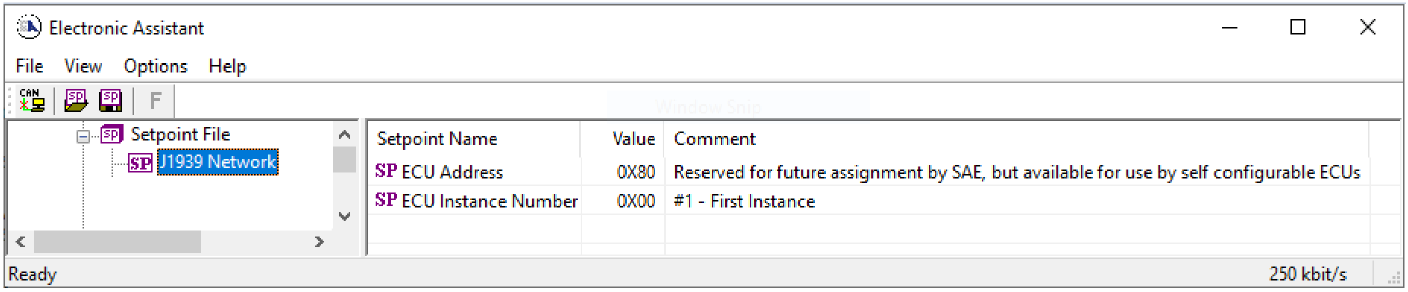


Figure 19 – Screen Capture of J1939 Setpoints

Table 20 – J1939 Network Setpoints

Name	**Range**	**Default**	**Notes**
ECU Address	0x80	0-253	Preferred address for a self-configurable ECU
ECU Instance	0-7	0x00	Per J1939-81

If non-default values for the **"ECU Instance Number"** or **"ECU Address"** are used, they will be mirrored during a setpoint file flashing, and will only take effect once the entire file has been downloaded to the unit. After the setpoint flashing is complete, the unit will claim the new address and/or re-claim the address with the new NAME. If these setpoints are changing, 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.

4.3. LIN Common

The LIN Common Block is described in Section 1.3.1. The default setpoints are shown in the Table 3.

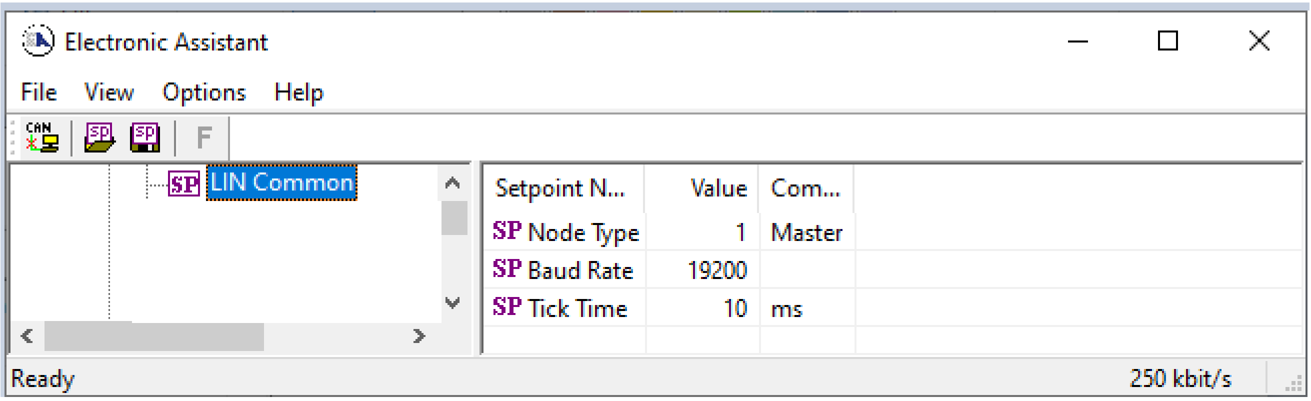
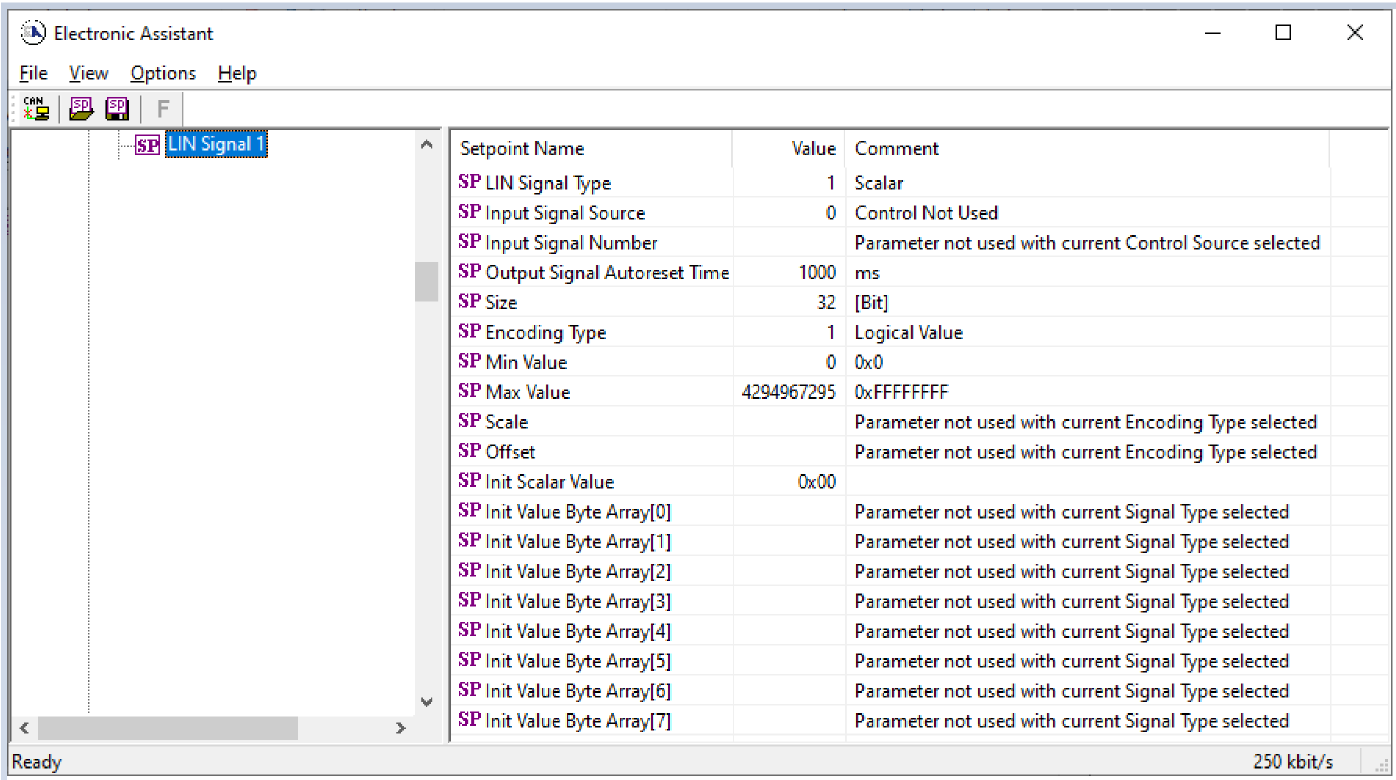


Figure 20 – Screen Capture of J1939 Setpoints

4.4. LIN Signal

The LIN Common Block is described in Section 1.3.2. The default setpoints are shown in the Table 4.

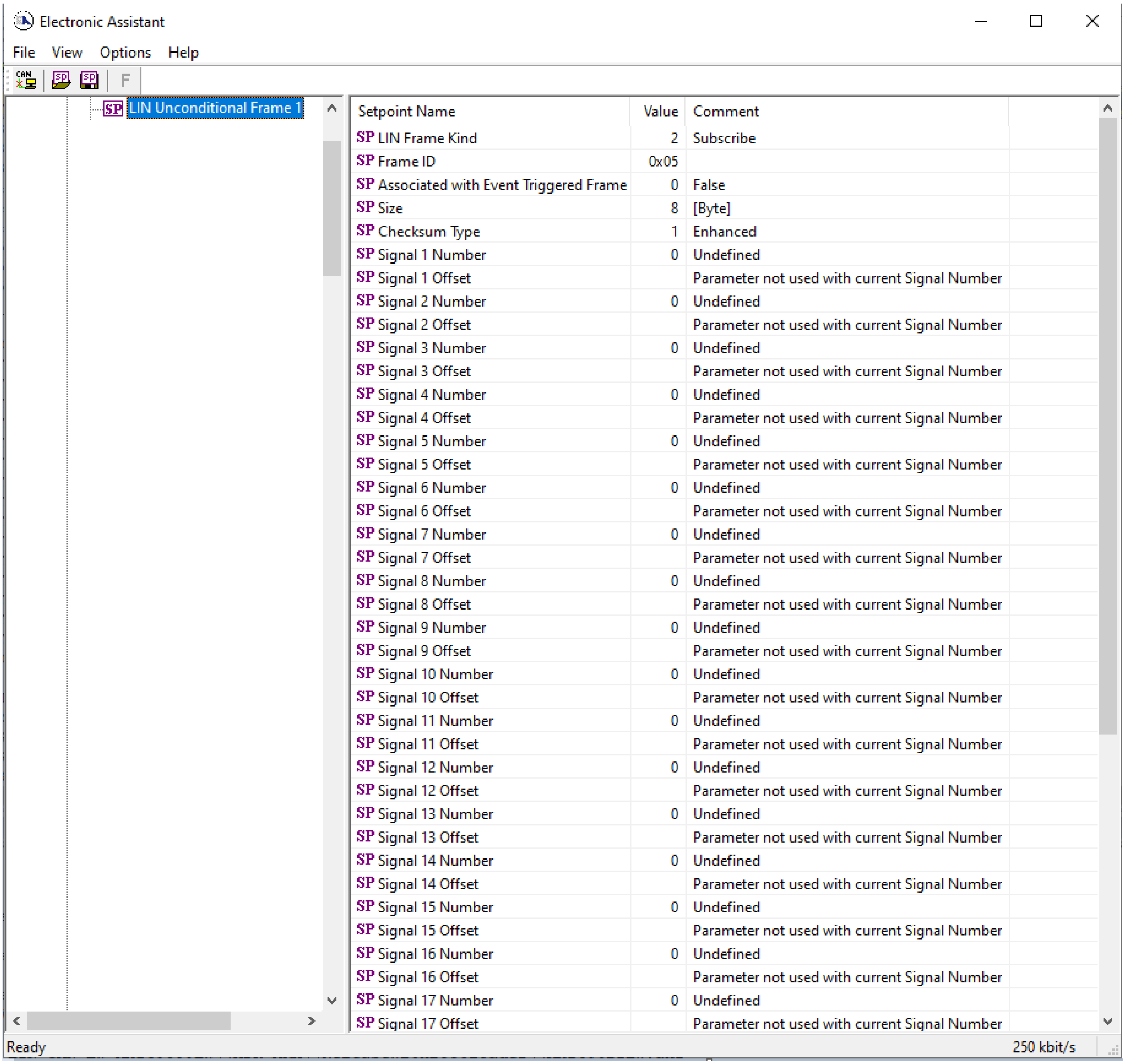


Setpoint Name	Value	Comment
SP LIN Signal Type	1	Scalar
SP Input Signal Source	0	Control Not Used
SP Input Signal Number		Parameter not used with current Control Source selected
SP Output Signal Autoreset Time	1000	ms
SP Size	32	[Bit]
SP Encoding Type	1	Logical Value
SP Min Value	0	0x0
SP Max Value	4294967295	0xFFFFFFFF
SP Scale		Parameter not used with current Encoding Type selected
SP Offset		Parameter not used with current Encoding Type selected
SP Init Scalar Value	0x00	
SP Init Value Byte Array[0]		Parameter not used with current Signal Type selected
SP Init Value Byte Array[1]		Parameter not used with current Signal Type selected
SP Init Value Byte Array[2]		Parameter not used with current Signal Type selected
SP Init Value Byte Array[3]		Parameter not used with current Signal Type selected
SP Init Value Byte Array[4]		Parameter not used with current Signal Type selected
SP Init Value Byte Array[5]		Parameter not used with current Signal Type selected
SP Init Value Byte Array[6]		Parameter not used with current Signal Type selected
SP Init Value Byte Array[7]		Parameter not used with current Signal Type selected

Figure 21 – Screen Capture of J1939 Setpoints

4.5. LIN Unconditional Frame

The LIN Common Block is described in Section 1.3.4. The default setpoints are shown in the Table 6.



Setpoint Name	Value	Comment
SP LIN Frame Kind	2	Subscribe
SP Frame ID	0x05	
SP Associated with Event Triggered Frame	0	False
SP Size	8	[Byte]
SP Checksum Type	1	Enhanced
SP Signal 1 Number	0	Undefined
SP Signal 1 Offset		Parameter not used with current Signal Number
SP Signal 2 Number	0	Undefined
SP Signal 2 Offset		Parameter not used with current Signal Number
SP Signal 3 Number	0	Undefined
SP Signal 3 Offset		Parameter not used with current Signal Number
SP Signal 4 Number	0	Undefined
SP Signal 4 Offset		Parameter not used with current Signal Number
SP Signal 5 Number	0	Undefined
SP Signal 5 Offset		Parameter not used with current Signal Number
SP Signal 6 Number	0	Undefined
SP Signal 6 Offset		Parameter not used with current Signal Number
SP Signal 7 Number	0	Undefined
SP Signal 7 Offset		Parameter not used with current Signal Number
SP Signal 8 Number	0	Undefined
SP Signal 8 Offset		Parameter not used with current Signal Number
SP Signal 9 Number	0	Undefined
SP Signal 9 Offset		Parameter not used with current Signal Number
SP Signal 10 Number	0	Undefined
SP Signal 10 Offset		Parameter not used with current Signal Number
SP Signal 11 Number	0	Undefined
SP Signal 11 Offset		Parameter not used with current Signal Number
SP Signal 12 Number	0	Undefined
SP Signal 12 Offset		Parameter not used with current Signal Number
SP Signal 13 Number	0	Undefined
SP Signal 13 Offset		Parameter not used with current Signal Number
SP Signal 14 Number	0	Undefined
SP Signal 14 Offset		Parameter not used with current Signal Number
SP Signal 15 Number	0	Undefined
SP Signal 15 Offset		Parameter not used with current Signal Number
SP Signal 16 Number	0	Undefined
SP Signal 16 Offset		Parameter not used with current Signal Number
SP Signal 17 Number	0	Undefined
SP Signal 17 Offset		Parameter not used with current Signal Number

Figure 22 – Screen Capture of J1939 Setpoints

4.6. LIN Event Triggered Frame

The LIN Event Triggered Frame Block is described in Section 1.3.5. The default setpoints are shown in the Table 7.

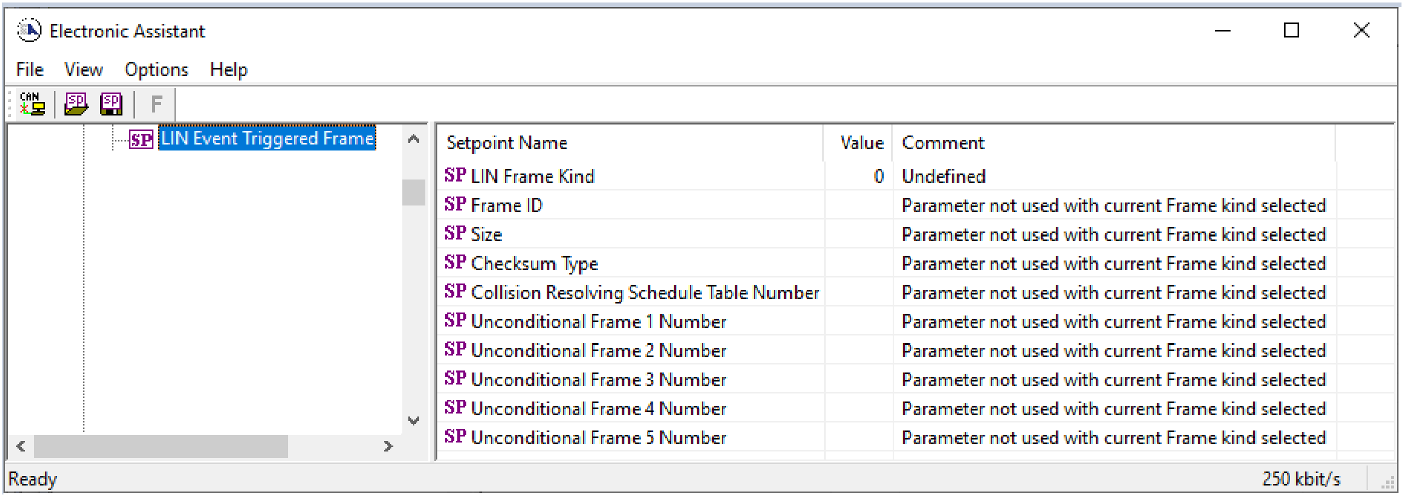


Figure 23 – Screen Capture of J1939 Setpoints

4.7. LIN Sporadic Frame

The LIN Sporadic Frame Block is described in Section 1.3.6. The default setpoints are shown in the Table 8.

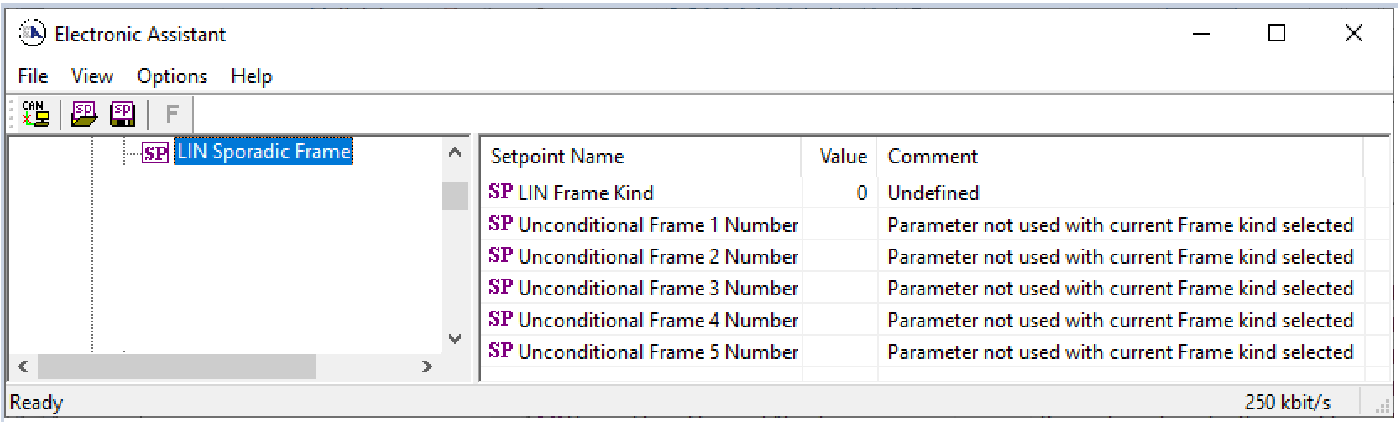
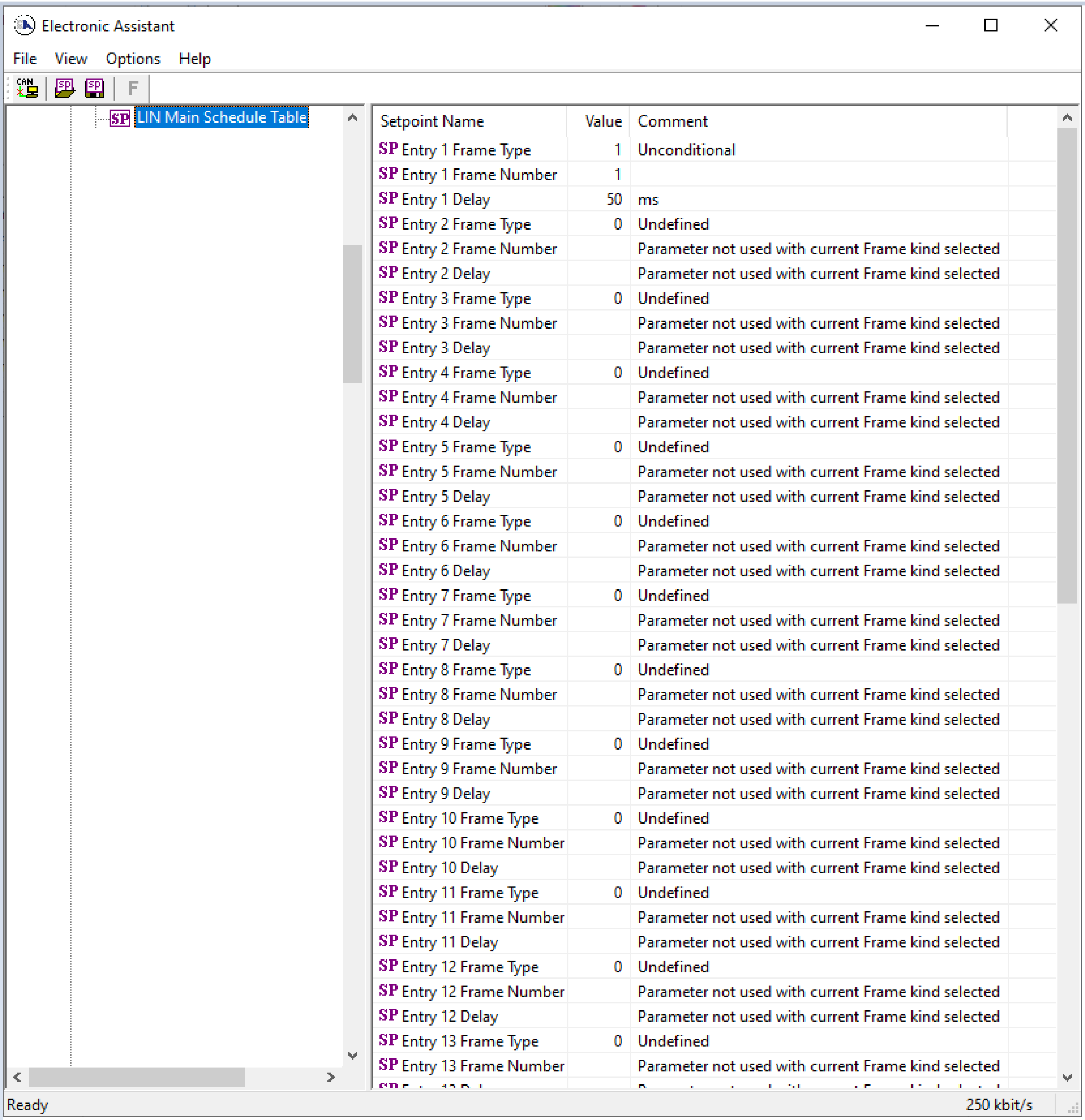


Figure 24 – Screen Capture of J1939 Setpoints

4.8. LIN Main Schedule Table

The LIN Main Schedule Table Block is described in Section 1.3.7. The default setpoints are shown in the Table 9.

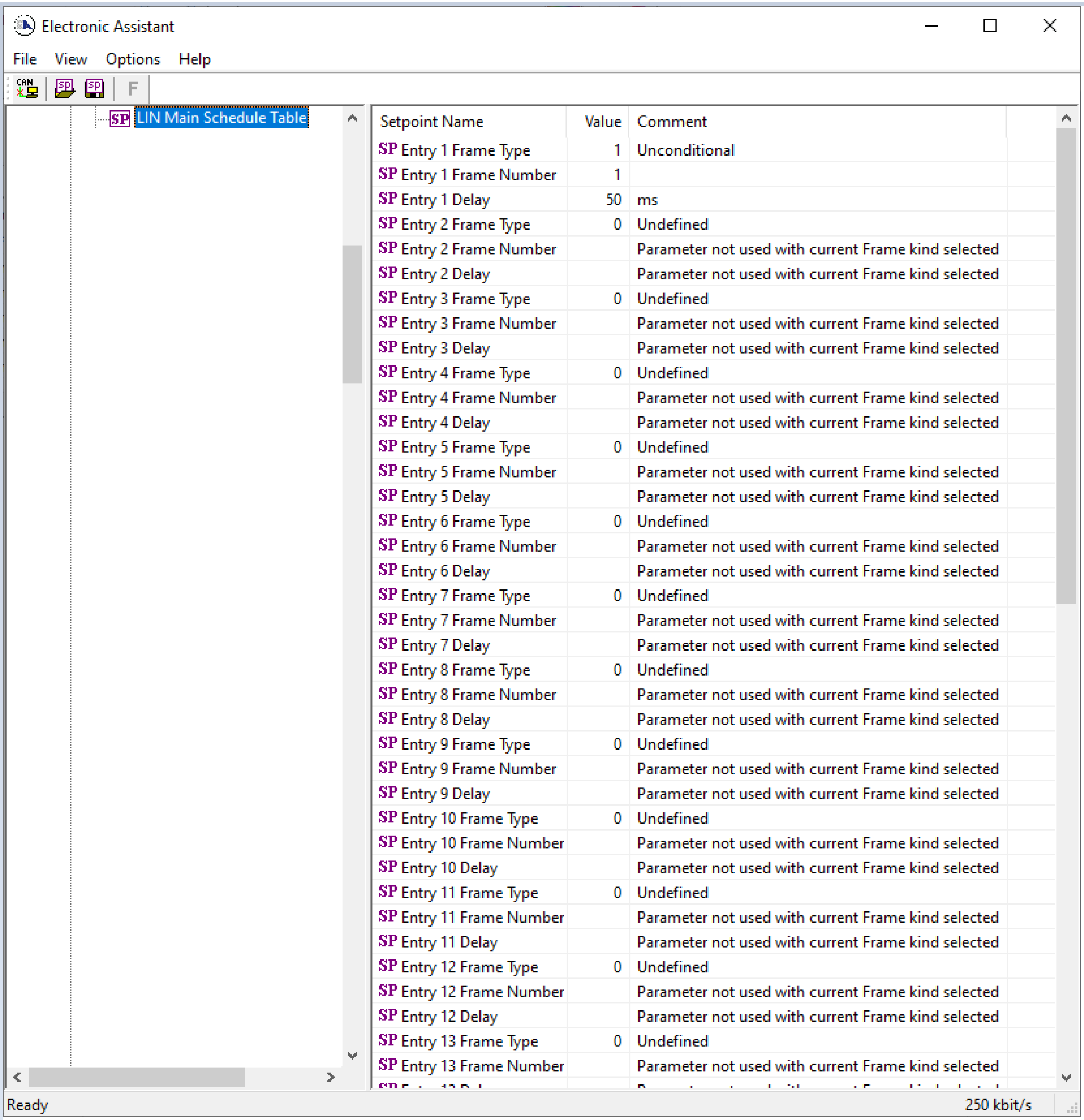


Setpoint Name	Value	Comment
SP Entry 1 Frame Type	1	Unconditional
SP Entry 1 Frame Number	1	
SP Entry 1 Delay	50	ms
SP Entry 2 Frame Type	0	Undefined
SP Entry 2 Frame Number		Parameter not used with current Frame kind selected
SP Entry 2 Delay		Parameter not used with current Frame kind selected
SP Entry 3 Frame Type	0	Undefined
SP Entry 3 Frame Number		Parameter not used with current Frame kind selected
SP Entry 3 Delay		Parameter not used with current Frame kind selected
SP Entry 4 Frame Type	0	Undefined
SP Entry 4 Frame Number		Parameter not used with current Frame kind selected
SP Entry 4 Delay		Parameter not used with current Frame kind selected
SP Entry 5 Frame Type	0	Undefined
SP Entry 5 Frame Number		Parameter not used with current Frame kind selected
SP Entry 5 Delay		Parameter not used with current Frame kind selected
SP Entry 6 Frame Type	0	Undefined
SP Entry 6 Frame Number		Parameter not used with current Frame kind selected
SP Entry 6 Delay		Parameter not used with current Frame kind selected
SP Entry 7 Frame Type	0	Undefined
SP Entry 7 Frame Number		Parameter not used with current Frame kind selected
SP Entry 7 Delay		Parameter not used with current Frame kind selected
SP Entry 8 Frame Type	0	Undefined
SP Entry 8 Frame Number		Parameter not used with current Frame kind selected
SP Entry 8 Delay		Parameter not used with current Frame kind selected
SP Entry 9 Frame Type	0	Undefined
SP Entry 9 Frame Number		Parameter not used with current Frame kind selected
SP Entry 9 Delay		Parameter not used with current Frame kind selected
SP Entry 10 Frame Type	0	Undefined
SP Entry 10 Frame Number		Parameter not used with current Frame kind selected
SP Entry 10 Delay		Parameter not used with current Frame kind selected
SP Entry 11 Frame Type	0	Undefined
SP Entry 11 Frame Number		Parameter not used with current Frame kind selected
SP Entry 11 Delay		Parameter not used with current Frame kind selected
SP Entry 12 Frame Type	0	Undefined
SP Entry 12 Frame Number		Parameter not used with current Frame kind selected
SP Entry 12 Delay		Parameter not used with current Frame kind selected
SP Entry 13 Frame Type	0	Undefined
SP Entry 13 Frame Number		Parameter not used with current Frame kind selected

Figure 25 – Screen Capture of J1939 Setpoints

4.9. LIN Collision Schedule Table

The LIN Collision Schedule Table Block is described in Section 1.3.8. The default setpoints are shown in the Table 10.



Setpoint Name	Value	Comment
SP Entry 1 Frame Type	1	Unconditional
SP Entry 1 Frame Number	1	
SP Entry 1 Delay	50	ms
SP Entry 2 Frame Type	0	Undefined
SP Entry 2 Frame Number		Parameter not used with current Frame kind selected
SP Entry 2 Delay		Parameter not used with current Frame kind selected
SP Entry 3 Frame Type	0	Undefined
SP Entry 3 Frame Number		Parameter not used with current Frame kind selected
SP Entry 3 Delay		Parameter not used with current Frame kind selected
SP Entry 4 Frame Type	0	Undefined
SP Entry 4 Frame Number		Parameter not used with current Frame kind selected
SP Entry 4 Delay		Parameter not used with current Frame kind selected
SP Entry 5 Frame Type	0	Undefined
SP Entry 5 Frame Number		Parameter not used with current Frame kind selected
SP Entry 5 Delay		Parameter not used with current Frame kind selected
SP Entry 6 Frame Type	0	Undefined
SP Entry 6 Frame Number		Parameter not used with current Frame kind selected
SP Entry 6 Delay		Parameter not used with current Frame kind selected
SP Entry 7 Frame Type	0	Undefined
SP Entry 7 Frame Number		Parameter not used with current Frame kind selected
SP Entry 7 Delay		Parameter not used with current Frame kind selected
SP Entry 8 Frame Type	0	Undefined
SP Entry 8 Frame Number		Parameter not used with current Frame kind selected
SP Entry 8 Delay		Parameter not used with current Frame kind selected
SP Entry 9 Frame Type	0	Undefined
SP Entry 9 Frame Number		Parameter not used with current Frame kind selected
SP Entry 9 Delay		Parameter not used with current Frame kind selected
SP Entry 10 Frame Type	0	Undefined
SP Entry 10 Frame Number		Parameter not used with current Frame kind selected
SP Entry 10 Delay		Parameter not used with current Frame kind selected
SP Entry 11 Frame Type	0	Undefined
SP Entry 11 Frame Number		Parameter not used with current Frame kind selected
SP Entry 11 Delay		Parameter not used with current Frame kind selected
SP Entry 12 Frame Type	0	Undefined
SP Entry 12 Frame Number		Parameter not used with current Frame kind selected
SP Entry 12 Delay		Parameter not used with current Frame kind selected
SP Entry 13 Frame Type	0	Undefined
SP Entry 13 Frame Number		Parameter not used with current Frame kind selected

Figure 26 – Screen Capture of J1939 Setpoints

4.10. CAN Common

The CAN Common Block is described in Section 1.9.

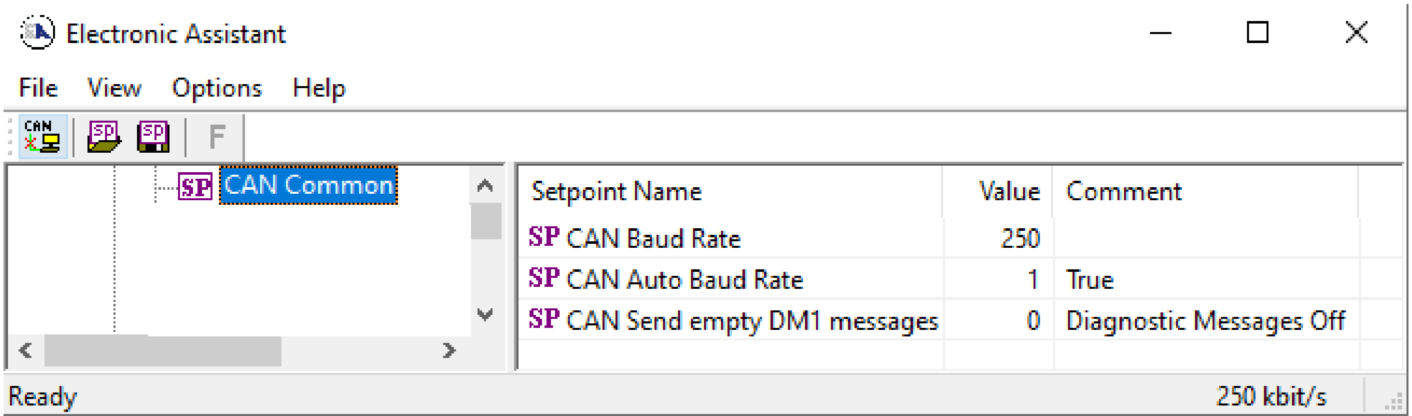


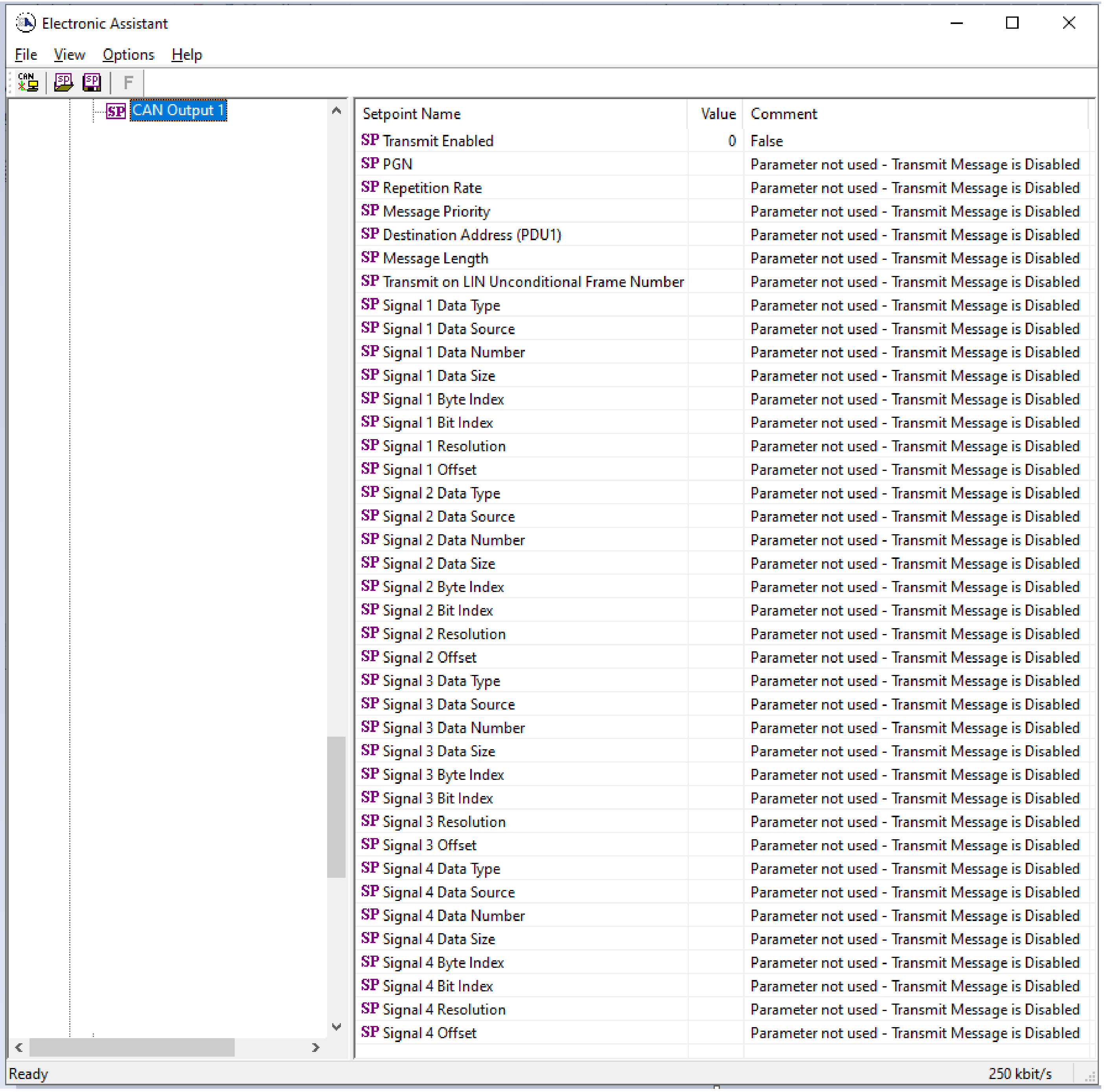
Figure 27 – Screen Capture of J1939 Setpoints

Table 21 – J1939 Network Setpoints

Name	Range	Default	Notes
CAN Baud Rate	125, 250, 500, 667, and 1000	250	Stores the current baud rate
CAN Auto Baud Rate	False/True	True	
CAN Send empty DM1 messages	0-2	0, Diagnostic Messages Off	

4.11. CAN Output Setpoints

CAN Transmit Message Function Block is presented in section 1.17. Please refer there for detailed information how these setpoints are used. "**Repetition Rate**" is 0ms by default, thus no message will be sent unless the "**Transmit on LIN Unconditional Frame Number**" is set to a non-zero value.



Setpoint Name	Value	Comment
SP Transmit Enabled	0	False
SP PGN		Parameter not used - Transmit Message is Disabled
SP Repetition Rate		Parameter not used - Transmit Message is Disabled
SP Message Priority		Parameter not used - Transmit Message is Disabled
SP Destination Address (PDU1)		Parameter not used - Transmit Message is Disabled
SP Message Length		Parameter not used - Transmit Message is Disabled
SP Transmit on LIN Unconditional Frame Number		Parameter not used - Transmit Message is Disabled
SP Signal 1 Data Type		Parameter not used - Transmit Message is Disabled
SP Signal 1 Data Source		Parameter not used - Transmit Message is Disabled
SP Signal 1 Data Number		Parameter not used - Transmit Message is Disabled
SP Signal 1 Data Size		Parameter not used - Transmit Message is Disabled
SP Signal 1 Byte Index		Parameter not used - Transmit Message is Disabled
SP Signal 1 Bit Index		Parameter not used - Transmit Message is Disabled
SP Signal 1 Resolution		Parameter not used - Transmit Message is Disabled
SP Signal 1 Offset		Parameter not used - Transmit Message is Disabled
SP Signal 2 Data Type		Parameter not used - Transmit Message is Disabled
SP Signal 2 Data Source		Parameter not used - Transmit Message is Disabled
SP Signal 2 Data Number		Parameter not used - Transmit Message is Disabled
SP Signal 2 Data Size		Parameter not used - Transmit Message is Disabled
SP Signal 2 Byte Index		Parameter not used - Transmit Message is Disabled
SP Signal 2 Bit Index		Parameter not used - Transmit Message is Disabled
SP Signal 2 Resolution		Parameter not used - Transmit Message is Disabled
SP Signal 2 Offset		Parameter not used - Transmit Message is Disabled
SP Signal 3 Data Type		Parameter not used - Transmit Message is Disabled
SP Signal 3 Data Source		Parameter not used - Transmit Message is Disabled
SP Signal 3 Data Number		Parameter not used - Transmit Message is Disabled
SP Signal 3 Data Size		Parameter not used - Transmit Message is Disabled
SP Signal 3 Byte Index		Parameter not used - Transmit Message is Disabled
SP Signal 3 Bit Index		Parameter not used - Transmit Message is Disabled
SP Signal 3 Resolution		Parameter not used - Transmit Message is Disabled
SP Signal 3 Offset		Parameter not used - Transmit Message is Disabled
SP Signal 4 Data Type		Parameter not used - Transmit Message is Disabled
SP Signal 4 Data Source		Parameter not used - Transmit Message is Disabled
SP Signal 4 Data Number		Parameter not used - Transmit Message is Disabled
SP Signal 4 Data Size		Parameter not used - Transmit Message is Disabled
SP Signal 4 Byte Index		Parameter not used - Transmit Message is Disabled
SP Signal 4 Bit Index		Parameter not used - Transmit Message is Disabled
SP Signal 4 Resolution		Parameter not used - Transmit Message is Disabled
SP Signal 4 Offset		Parameter not used - Transmit Message is Disabled

Figure 28 – Screen Capture of CAN Transmit Message Setpoints

Table 22 – CAN Transmit Message Setpoints

Name	Range	Default	Notes
Transmit Enabled	Drop List	0, False	
PGN	0xff00 … 0xffff	Different for each	See section 1.17.1
Repetition Rate	0 … 65000 ms	0ms	0ms disables transmit
Message Priority	0...7	6	Proprietary B Priority
Destination Address (PDU1)	0...255	255	Not used by default
Message Length	1-8	8	
Transmit on LIN Unconditional Frame Number	0-130	0	See section 1.17.1
Signal X Data Type	Drop List	0, Undefined	
Signal X Data Source	Drop List	Different for each	See Table 1
Signal X Data Number	Drop List	Different for each	See 1.17.2
Signal X Data Size	Drop List	2 bytes	
Signal X Byte Index	0-7	0	
Signal X Bit Index	0-7	0	
Signal X Resolution	-100000.0 to 100000	1/bits	
Signal X Offset	-10000 to 10000	0.0	

4.12. CAN Input Setpoints

The CAN Receive Block is defined in section 1.18. Please refer there for detailed information about how these setpoints are used. “**Receive Message Timeout**” is set to 0ms by default. To enable Receive message set “**Receive Message Timeout**” that differs from zero.

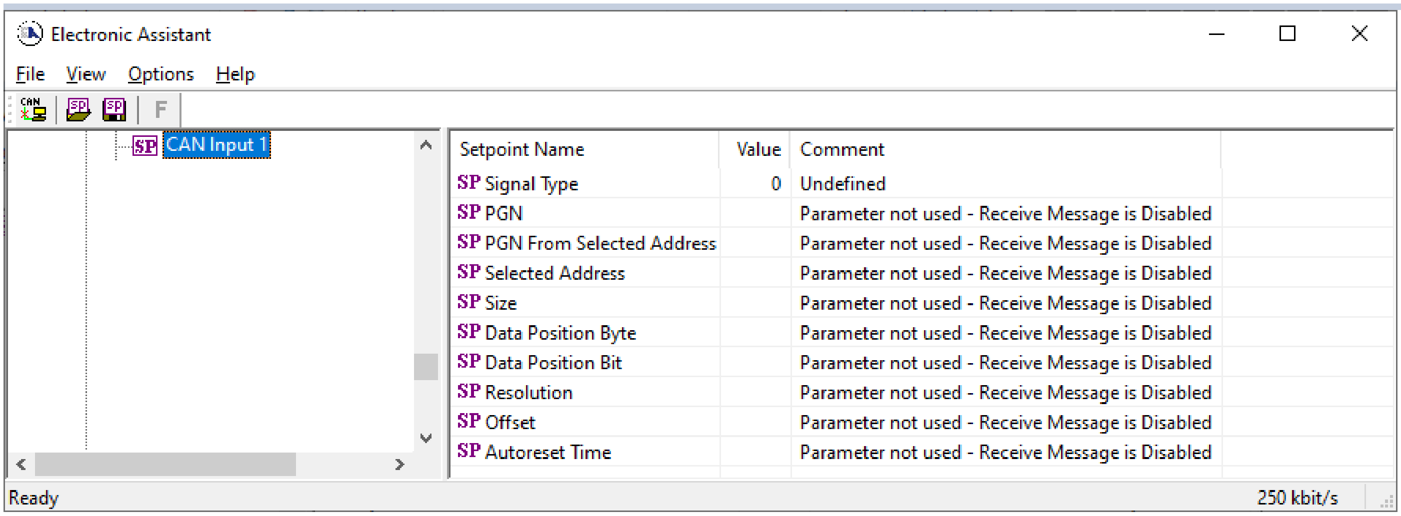


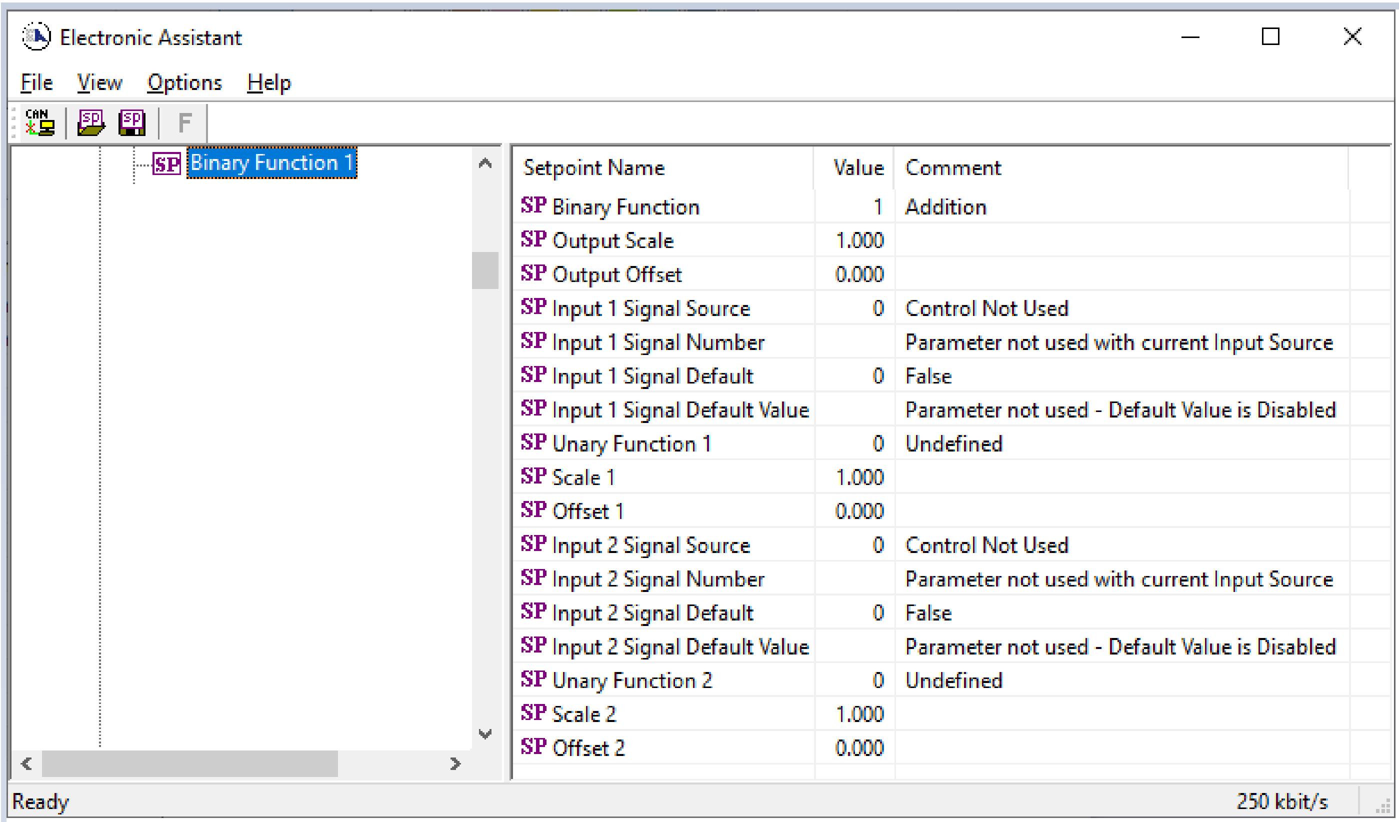
Figure 29 – Screen Capture of CAN Receive Message Setpoints

Table 23 – CAN Receive Setpoints

Name	Range	Default	Notes
Signal Type	Drop List	0, Undefined	
PGN	0 to 0x3FFFF	Different for each	
PGN From Selected Address	False/True	False	
Selected Address	0 to 255	254 (0xFE, Null Addr)	
Size	0 to 32	8 bit	
Data Position Byte	0-7	0	
Data Position Bit	0-7	0	
Resolution	-100000.0 to 100000	0.001	
Offset	-10000 to 10000	0.0	
Autoreset Time	0 to 60 000 ms	0ms	

4.13. Binary Function Block

The Binary Function Block is described in Section 1.4. The default setpoints are shown in the Table 13.



Setpoint Name	Value	Comment
SP Binary Function	1	Addition
SP Output Scale	1.000	
SP Output Offset	0.000	
SP Input 1 Signal Source	0	Control Not Used
SP Input 1 Signal Number		Parameter not used with current Input Source
SP Input 1 Signal Default	0	False
SP Input 1 Signal Default Value		Parameter not used - Default Value is Disabled
SP Unary Function 1	0	Undefined
SP Scale 1	1.000	
SP Offset 1	0.000	
SP Input 2 Signal Source	0	Control Not Used
SP Input 2 Signal Number		Parameter not used with current Input Source
SP Input 2 Signal Default	0	False
SP Input 2 Signal Default Value		Parameter not used - Default Value is Disabled
SP Unary Function 2	0	Undefined
SP Scale 2	1.000	
SP Offset 2	0.000	

Figure 30 – Screen Capture of J1939 Setpoints

4.14. Constant Data List

The Constant Data List Function Block allows the user to select values as desired for various logic block functions.

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 programmable to any value between +/. 1 000 000. The default values (shown in Figure 16) are arbitrary and should be configured by the user as appropriate for their application.

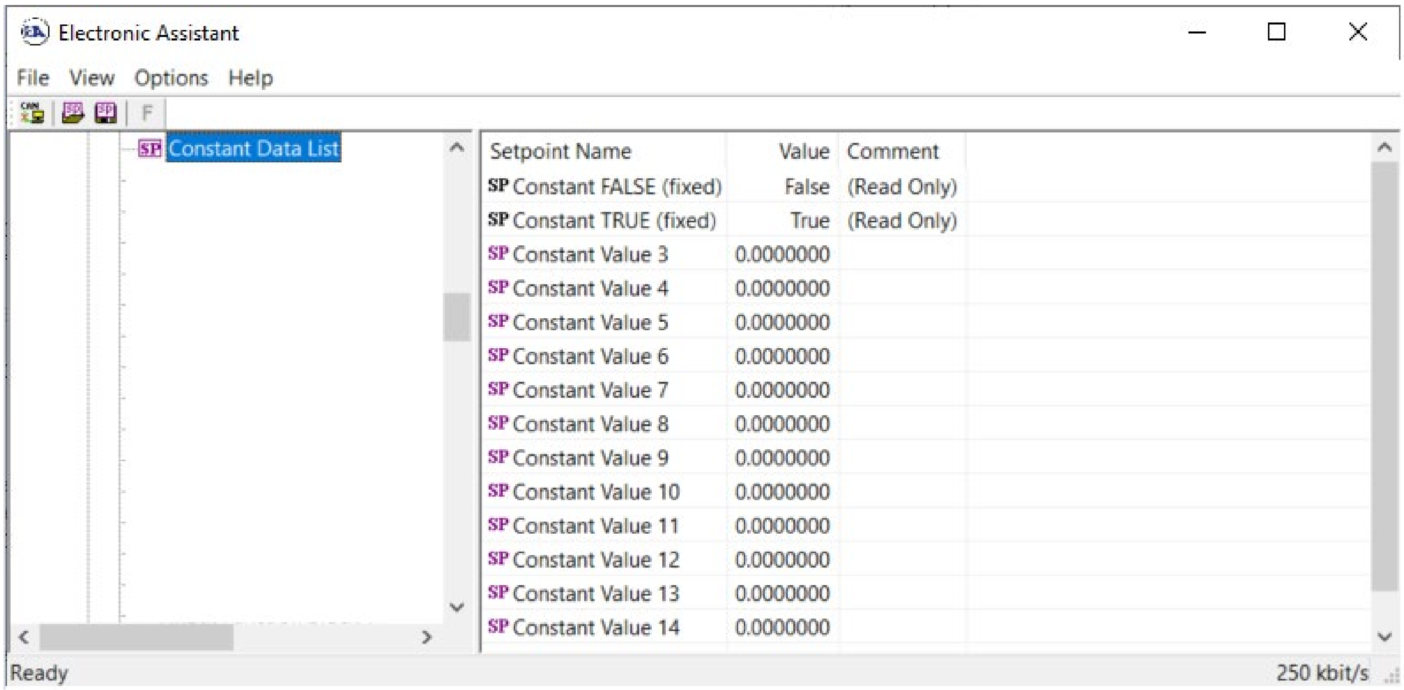


Figure 31 – Screen Capture of Constant Data List Setpoints

4.15. Diagnostics Blocks

There are 16 Diagnostics blocks that can be configured to monitor various parameters of the Controller. The Diagnostic Function Block is defined in section 1.9. Please refer there for detailed information on how these setpoints are used.

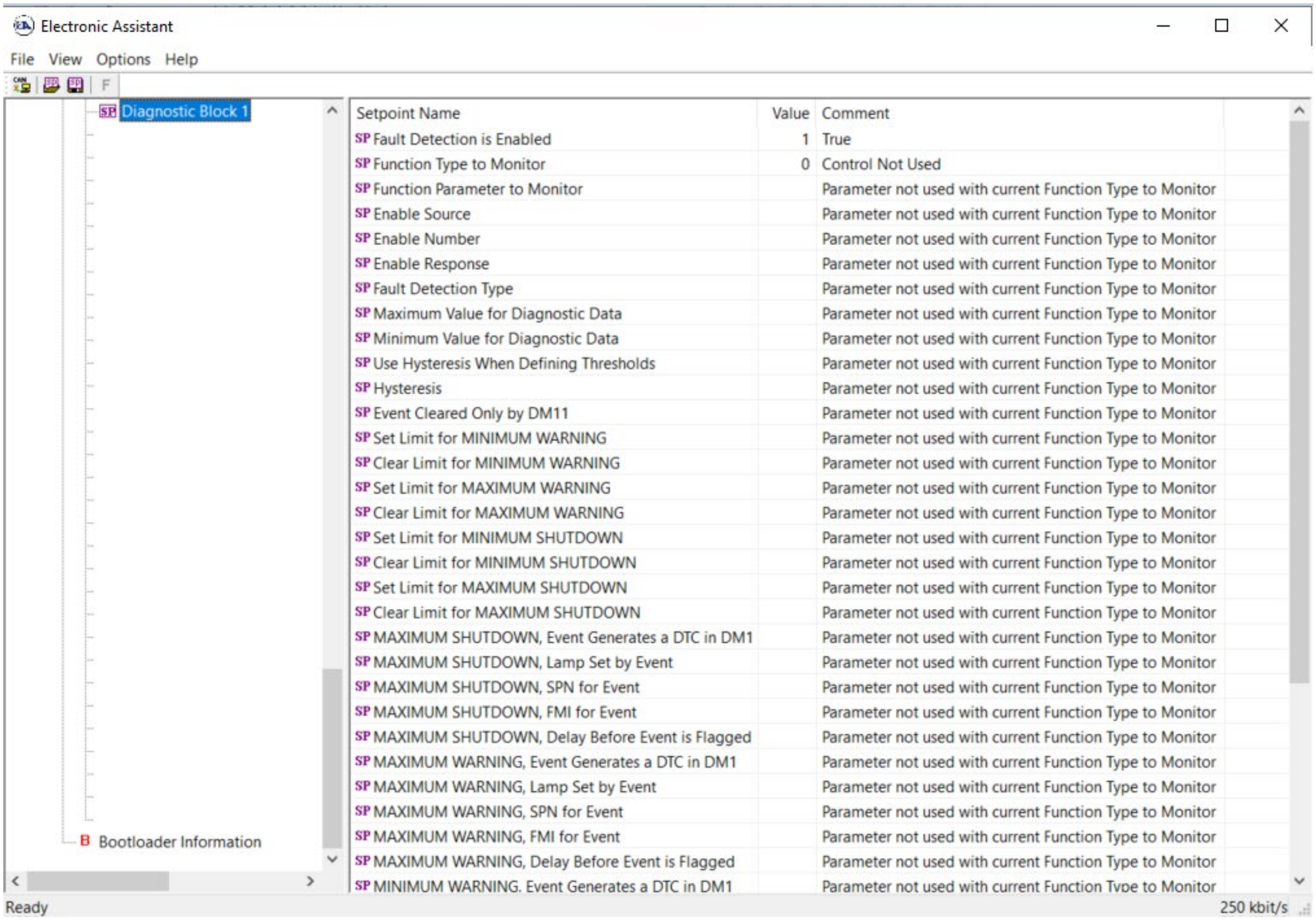


Figure 32 – Screen Capture of Diagnostic Block Setpoints

Table 24 – Diagnostic Block 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	0 – Min and Max Error	See section 1.9
Maximum Value for Diagnostic Data	Minimum Value for Diagnostic Data … $4.28e^9$	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 … Maximum Value for Diagnostics Data	4.8	
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 21

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 22
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 21
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 22
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 Event	Drop List	0 – Protect	See Table 21
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 22
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 21
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 22
MINIMUM SHUTDOWN, Delay Before Event is Flagged	0…60000 ms	1000	

4.16. DTC React Function Block

The DTC React function block is described in Section 1.16. The Figure below shows the DTC React function block setpoints. The Table below show the default values. Please note: *The setpoint "DTC React is Enabled" was changed to 1, True.*

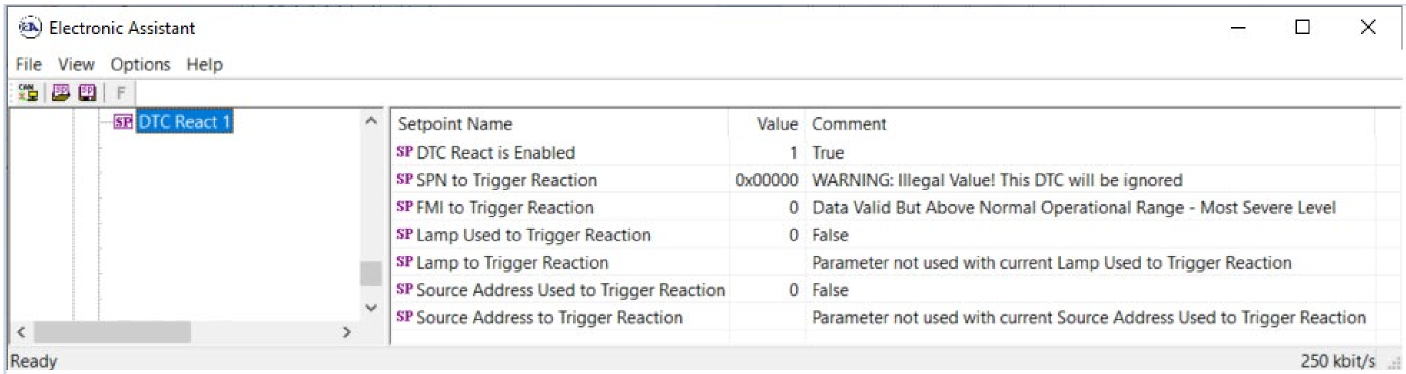


Figure 33 DTC React Setpoints

Table 25 – DTC React Setpoints

Name	Range	Default	Notes
DTC React is Enabled	Drop List	0, False	
SPN to Trigger Reaction	0x00 to 0x3FFFF	0	
FMI to Trigger Reaction	Drop List	0	
Lamp Used to Trigger Reaction	Drop list	0, False	
Lamp to Trigger Reaction	Drop List	0, Protect	
Source Address Used to Trigger Reaction	Drop list	0, False	
Source Address to Trigger Reaction	0x00 to 0xFF	0	

5. DIMENSIONS AND PINOUT

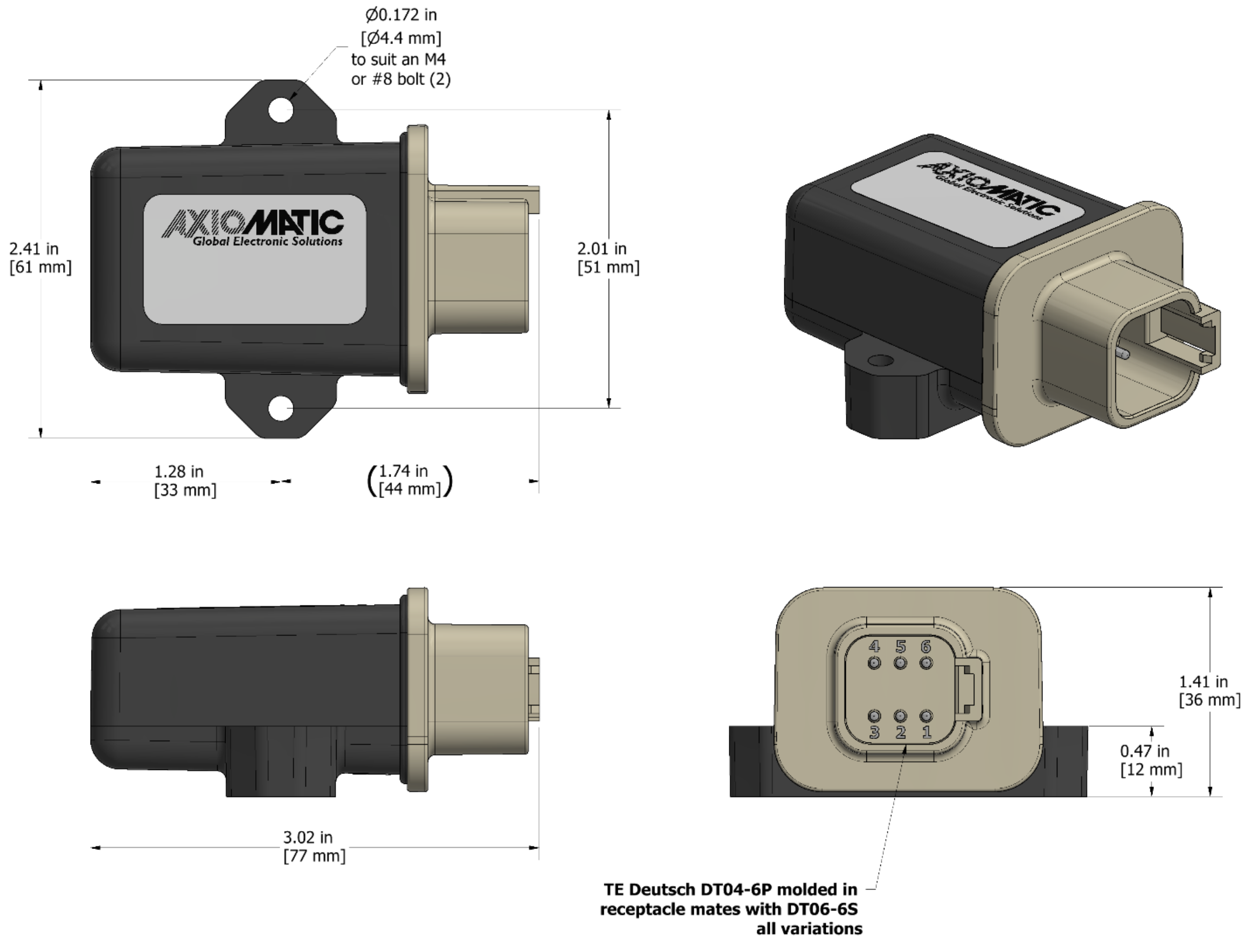


Figure 34 – Unit Dimensions

Table 26 – Connector Pinout

PIN	Description
1	CAN Shield
2	CAN Low
3	CAN High
4	LIN
5	Battery +
6	Battery -

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/.

Technical specifications are typical at nominal input voltage and 25 °C unless otherwise specified.

Power

Power Supply Input	12 or 24 VDC nominal (9 to 32 VDC range)
Quiescent Current	40 mA @ 12 V; 30 mA @ 24 V
Surge and Transient Protection	Provided
Reverse Polarity Protection	Provided up to 32 V
Undervoltage Protection	Undervoltage hardware shutdown at 4 V
Overvoltage Protection	Overvoltage hardware shutdown at 46 V

Communications

LIN Port	1 LIN (LIN 2.2A) Supported baud rate: 2.4 kbit/s to 20 kbit/s
CAN Port	1 CAN port (SAE J1939) Supported baud rate: 125 kbit/s, 250 kbit/s (default), 500 kbit/s, 667 kbit/s, and 1 Mbit/s with auto-baud-rate detection CANopen® model: AX140621

General Specifications

Microcontroller	STM32H725RGV3 32-bit, 1 MB flash program memory
User Interface	Axiomatic Electronic Assistant Kit - P/N: **AX070502** or **AX070506K**
Compliance	RoHS
Operating Conditions	-40 to 85 °C (-40 to 185 °F)
Storage Conditions	-40 to 85 °C (-40 to 185 °F)
Weight	0.15 lb. (0.068 kg)
Protection Rating	IP67
Enclosure	Plastic enclosure, Laser welded Nylon 6-6 with 30% glass fill 3.02 in. x 2.41 in. x 1.41 in. (77 mm x 61 mm x 36 mm) L x W x H includes the integral connector Refer to dimensional drawing.
Electrical Connections	1x integral 6-pin connector equivalent to TE Deutsch P/N: DT04-6P
Mating Connector	A mating plug kit is available from Axiomatic under P/N: **AX070119** (includes 1x DT06-6S, 1x W6S, and 6x 0462-201-16141)

7. VERSION HISTORY

Version	Date	Author	Modifications
-	May. 5, 2025	Dmytro Tsebrii	Initial draft
1.0.0	Oct. 15, 2025	Dmytro Tsebrii	Prepared for initial release
1.0.1	Oct. 16, 2025	M Ejaz	Updated technical specifications Marketing review

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