User Manual UMAX180800
Version 2A
Firmware 2.xx
EA 5.16.139.0+

USER MANUAL

20 Thermocouple, 2 RTD, 4 Inputs, 6 Relays
Dual CAN Controller
P/N: AX180800

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

ARP	Address Resolution Protocol
ASCII	American Standard Code for Information Interchange
AWG	American wire gauge
CAN	Controller Area Network
CE	Conformité Européenne (European Conformity)
CMOS	Complementary metal-oxide-semiconductor
DC	Direct Current
DIN	German Institute for Standardization
DM	Diagnostic message. Defined in J1939/73 standard
EA	Axiomatic Electronic Assistant. (PC application software from Axiomatic)
ECU	Electronic control unit
EEPROM	Electrically Erasable Programmable Read-Only Memory
EMI	Electromagnetic Interference
EN	European Standard
GPL	General Public License
ICMP	Internet Control Message Protocol
ID	Identifier
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol or Ingress Protection (for enclosure)
ISO	International Organization for Standardization
LAN	Local Area Network
LED	Light-emitting diode
LoZ	Low resistance
LSB	Less Significant Byte
MAC	Media Access Control (address)
MDIX	Medium Dependent Interface Crossover (MDI-X)
PC	Personal Computer
PGN	Parameter Group Number. Defined in J1939/73 standard
PHY	Physical Layer Transceiver (Ethernet chip)
P/N	Part Number
PWM	Pulse-width modulation
RoHS	Restriction of Hazardous Substances
RTOS	Real-Time Operating System
SAE J1939	CAN-based higher-level protocol designed and supported by the Society of automobile Engineers (SAE)
S/N	Serial Number
TBD	To be determined
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UL	Underwriters Laboratories (safety organization)
USB	Universal Serial Bus
VDC	Volt Direct Current
UTP	Unshielded twisted pair

TABLE OF CONTENTS

1 INTRODUCTION

The following user manual describes architecture, functionality, configuration parameters and flashing instructions for the 20 Thermocouple, 2 RTD, 4 Inputs, 6 Relays Dual CAN Controller with SAE J1939 CAN and Modbus TCP/IP Ethernet communication links. It also contains controller technical specifications and installation instructions to help users build a custom solution on the base of this controller.

The user should check whether the application firmware installed in the controller is covered by this user manual. It can be done through CAN bus using Axiomatic Electronic Assistant (EA) software or using Ethernet Modbus TCP/IP link.

The user manual is valid for application firmware with the same major version number as the user manual. For example, this user manual is valid for any converter application firmware V1.xx. Updates specific to the user manual are done by adding letters: A, B, …, Z to the user manual version number.

It is assumed, that the user is familiar with Modbus and J1939 CAN groups of standards. The terminology from these standards is widely used in this manual.

2 CONTROLLER DESCRIPTION

The controller is designed to convert physical signals from bipolar and universal inputs into J1939 CAN signals and input register data for the Modbus TCP interface. The universal inputs accept voltage, current, frequency, PWM duty cycle, and discrete voltage levels.

The J1939 CAN network can operate at standard 250 and 500 kbit/s and non-standard 667kbit/s and 1Mbit/s baud rates. The required baud rate is detected automatically upon connection to the CAN network.

The Modbus TCP/IP interface runs on a standard 10/100 Mbit/s Ethernet link providing up to 5 simultaneous client connections.

The controller can be configured through a set of configuration parameters over CAN or Ethernet interface to fit the user-specific application requirements.

2.1 Hardware Block Diagram

The controller contains 20 thermocouple, 2 RTD, and 4 universal inputs, 6 relays, two CAN interfaces and one Ethernet port, and a protected power supply. An embedded 32-bit microcontroller provides necessary processing power to the controller.

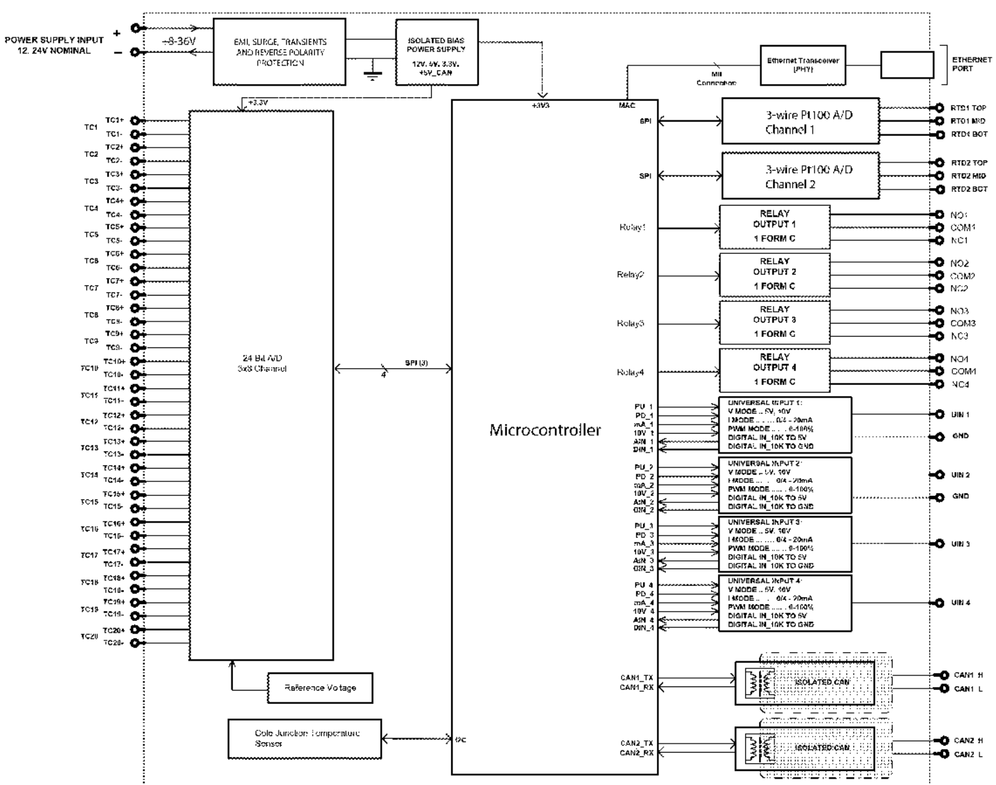


Figure 1. The Controller Hardware Block Diagram

The controller has a wide range of protection features including a transient and reverse polarity protection, see TECHNICAL SPECIFICATIONS section.

2.2 Software Organization

The controller belongs to a family of Axiomatic smart controllers with configurable internal architecture. This architecture allows building of a signal converting algorithm based on a set of predefined internal configurable function blocks without the need of a custom software.

The user can configure the controller internal structure and individual function blocks using PC-based Axiomatic Electronic Assistant (EA) software through CAN interface, without disconnecting the converter from the user system. Alternatively, the user can configure the controller through the Modbus link.

The controller application firmware can be updated through CAN interface in the field using EA, see FLASHING NEW FIRMWARE section. Updating firmware over Modbus is currently not implemented.

2.3 CAN Interface

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

Table 1. 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 controller 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 controller 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 controller 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 controller can receive and transmit application specific PGNs. All application-specific PGNs are user-programmable.
	J1939/73 – Application Layer – Diagnostics. Rev. FEB 2010

ISO/OSI Network Model Layer	J1939 Standard
	Memory access protocol (MAP) support. DM14, DM15, DM16 are used by EA to program configuration parameters. DM13 support is provided to temporarily suppress transmission of application specific PGNs.

2.3.1 CAN Baud Rate

The controller can operate at J1939 standard 250 and 500 kbit/s baud rates. It can also run at 667kbit/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 controller power-up.

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

2.3.2 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 2. 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	126 (IO Controller, Axiomatic proprietary)	No
Function Instance	5 bit	20 (AX180800, Axiomatic 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 controller *ECU Instance* using EA to accommodate multiple signal input controllers on the same CAN network.

The controller 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 EA.

2.3.3 Slew Rate Control

2.4 The controller has an ability to adjust the CAN transceiver slew rate for better performance on the CAN physical network, see Miscellaneous

The Miscellaneous function block contains various parameters that affect the general diagnostic performance of the ECU.

The **Undervoltage Threshold**, **Overvoltage Threshold**, and **Shutdown Temperature** setpoints are used to set the limits for when their respective diagnostic messages are triggered.

Lastly, the **CAN Diagnostic Setting1** and **CAN Diagnostic Setting2** parameters are used to control all diagnostics with one general setting for CAN Interface 1 and CAN Interface 2 respectively. This can be used to disable diagnostics entirely, only transmit messages without a blank SPN, or transmit diagnostic messages normally.

2.5 Diagnostics

The Diagnostic function block includes 120 faults, each representing a diagnostic message that the ECU can produce. Each Universal Input has a Voltage Out of Range Low and Voltage Out of Range High Faut. Each RTD has Temperature Out of Range Low and Temperature Out of Range High, High and Low Shutdown Faults. Each Thermocouple Input has Temperature Out of Range Low and Temperature Out of Range High, High and Low Shutdown, and Open Circuit Faults. The remaining faults cover VPS Overvoltage and Undervoltage, Overtemperature, and other faults.

If and only if the **Event Generates a DTC in DM1** parameter 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 4-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 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 Controller will send the DM1 message every one second, regardless of whether there are any active faults, as recommended by the standard. While there are no active DTCs, the Controller will send the “No Active Faults” message. If a previously active DTC becomes inactive, 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 on 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 broadcast 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.

The Diagnostic function block has a setpoint **Event Cleared Only by DM11**. By default, this is 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 change, 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. In those cases, the FMI in the setpoint reflects that of the low-end condition, and the FMI used by the high fault will be determined per Table 31. If the FMI is changed, the OC of the associate error log is automatically reset to zero.

Table 31. Low Fault FMI versus High Fault FMI

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

If the FMI used is anything other than one of those in Table 31, then both the low and the high faults will be assigned the same FMI. This condition should be avoided, as the log will still use different OC for the two types of faults, even though they will be reported the same in the DTC.

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 the Diagnostic function block. If the fault has remained present during the delay time, then the controller will set the DTC to active, and it 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.

2.6 J1939 Network function block for further details.

2.6.1 Network Bus Terminating Resistors

The controller 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 most CAN transceivers to operate properly.

2.7 Modbus TCP Interface

The Modbus TCP/IP interface[1] runs over a standard 10/100 Mbit/s Ethernet link. The controller is presented as a slave device (a server) on the Modbus network. It supports with up to 8 simultaneous client connections from master devices.

[1]The interface is compliant with:

- MODBUS Messaging on TCP/IP Implementation Guide V1.0b. Modbus Organization. October 24, 2006, 46p.
- MODBUS Application Protocol Specification V1.1b3. Modbus Organization. April 26, 2012, 50p.

The following Modbus functions are supported by the controller.

Table 3. Modbus Functions Supported by the Controller

Name	Function Code/Subcode	Description
Read Discrete Inputs	2	Reads values of the universal inputs when they are in the discrete voltage level mode
Read Input Registers	4	Reads values of the universal inputs
Read Holding Registers	3	Reads one or several configuration parameters
Write Single Register	6	Writes a configuration parameter
Write Multiple Registers	16	Writes one or several configuration parameters
Read/Write Multiple Registers	23	Writes and then reads configuration parameters
Encapsulated Interface Transport	43/14	Reads Device Identification

The Unit Identifier in the Modbus TCP header is ignored.

Floating-point variables are presented in a standard IEEE 754 single-precision 32-bit format, most significant word first. Double-word 32-bit integers are also presented with the most significant word first.

Reading and writing operations on variables occupying more than one word (a 16-bit Modbus register) are buffered. The buffering is made transparent to the user. However, it should be taken into consideration that writing to a non-volatile memory is not performed until all registers assigned to the variable are written. The writing operation should be performed without overlapping (writing to the same register twice) and without breaking the writing operation sequence with a reading operation or a writing operation to a different variable.

The Modbus functions “Write Multiple Registers” and “Read/Write Multiple Registers”, when they include all registers assigned to a variable in one function call, meet the abovementioned writing requirements.

The Modbus writing operations are subject to a validity check. If a configuration parameter value is not in a valid range, the Modbus operation will succeed, but the configuration parameter will not be written.

The following device identification information can be read using the Encapsulated Interface Transport 43/14 function.

Table 4. Modbus Device Identification

Object ID	Object Name	Description
0x00	VendorName	“Axiomatic”
0x01	ProductCode	Controller P/N. “AX180800”
0x02	MajorMinorRevision	Current firmware version. For example, “V1.00”

Object ID	Object Name	Description
0x03	VendorUrl	“www.axiomatic.com”
0x04	ProductName	“IO Controller”
0x05	ModelName	Same as ProductCode. Controller P/N. “AX180800”
0x06	UserApplicationName	Firmware description. Depends on the firmware version. For V1.xx: “20 Thermocouple, 2 RTD, 4 Inputs, 6 Relays Dual CAN Controller”
0x80	SerialNumber	Private Object. Controller S/N. For example, “0012020016”

All device identification objects are presented in ASCII strings.

2.8 Discovery Protocol

The controller supports an Axiomatic proprietary protocol that allows the discovery of Axiomatic controllers on a LAN by sending a global UDP request on port 35100[1].

[1] O. Bogush, "Ethernet to CAN Converter Discovery Protocol. CAN-ENET, AX140900, Project 15129. Document version: 1," Axiomatic Technologies Corporation, October 26, 2016.

Axiomatic provides a Windows console application `AxioDisc.exe` that can be used to discover the controller. The application shows the controller MAC address, IP address, web server port (not used), device port (Modbus port), device port type (TCP), the controller part number and serial number, see Figure 2.

The `AxioDisc.exe`. application is available upon request.

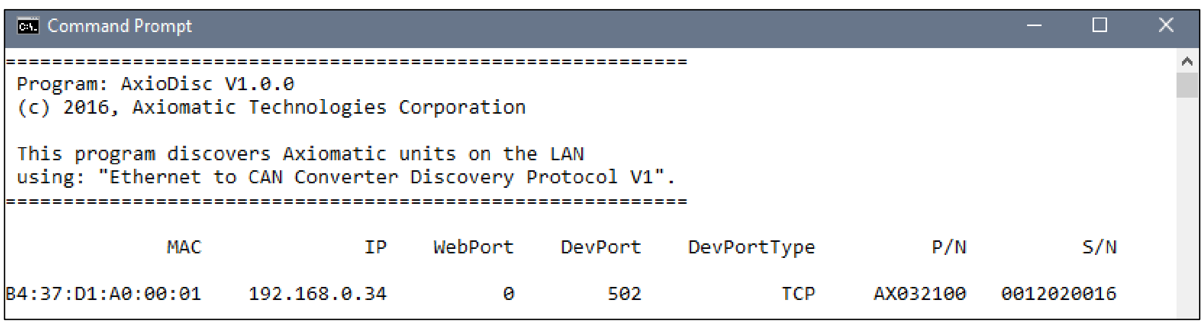


Figure 2. Discovery of the Controller on LAN Using `AxioDisc.exe` Application

2.9 Default Settings

The controller *Universal Inputs* are configured to input voltages in the 0…5V voltage range by default. These voltages can be read through Modbus interface.

The CAN output messages are not set up by default. They can be configured to accommodate user-specific application requirements, see, Section 4.2 in CONTROLLER CONFIGURATION section.

3 CONTROLLER LOGICAL STRUCTURE

The controller is internally organized as a set of function blocks, which can be individually configured and arbitrarily connected to achieve the required system functionality, see Figure 3.

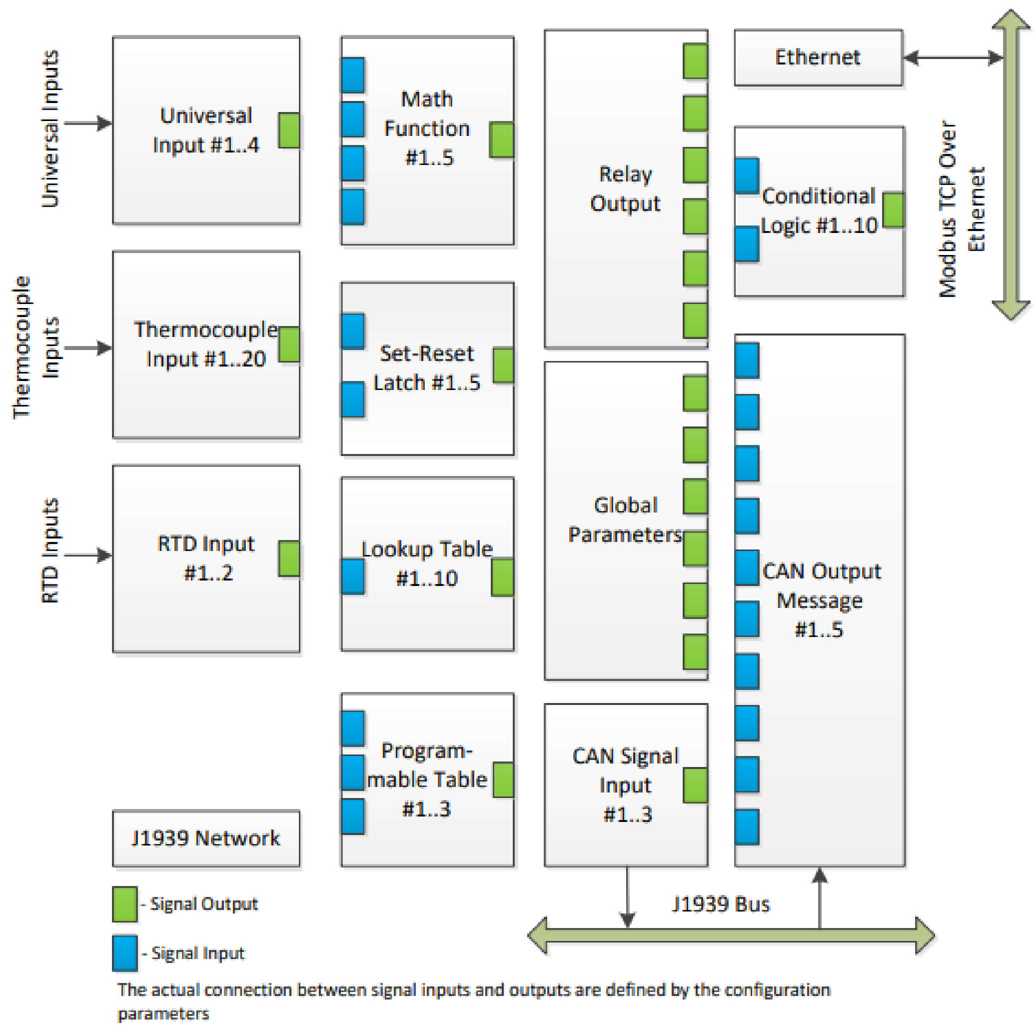


Figure 3. The Controller Logical Block Diagram

Each function block is absolutely independent and has its own set of configuration parameters, aka setpoints. The configuration parameters can be viewed and changed through CAN bus using Axiomatic Electronic Assistant (EA) software or over Modbus interface.

The *Universal Input* function block presents the controller physical input channels. This function block can measure multiple physical parameters.

The J1939 CAN interface is presented by the *CAN Input Signal, CAN Output Message* and *J1939 Network* function blocks. The *CAN Input Signal* functional blocks are used to receive CAN signals transmitted on the CAN bus. They have one signal output, which is updated once the signal is received. The *CAN Output Message* function blocks are used to transmit

CAN signals on the CAN bus. Each CAN message can hold up to ten individual CAN output signals, receiving data from their own signal inputs.

The Modbus interface is presented by the *Ethernet* function block that contains Modbus TCP/IP network settings.

For data processing, when required, there are *Math, Set-Reset Latch, Lookup Table Conditional Logic, and Programmable Logic* function blocks. They perform a unique set of functions that is explained in respective part of the Section 3.

The converter also has a *Global Parameters* function block containing four constant output signals and other auxiliary outputs.

3.1 Function Block Signals

The controller function blocks can communicate with each other through internal signal inputs and outputs. Each signal input of one function block can be connected to any signal output of another function block using an appropriate configuration parameter. There is no limitation on the number of signal inputs connected to one signal output.

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.

Function block signals can be “Undefined”, “Discrete” or “Continuous”. The “Undefined” signal type is reserved for a disconnected signal source or a no-signal transient condition. The “Discrete” and “Continuous” signal types are used to communicate discrete and continuous signals, respectively.

Discrete signals present data with a finite number of states. They are stored in four-byte unsigned integer variables that can present any state in the 0…0xFFFFFFFF range.

Continuous signals present continuous data, usually physical parameters. They are stored in single-precision floating-point variables. The continuous signals are not normalized and usually present data in physical units.

When a discrete signal output is connected to a continuous signal input, the discrete signal is converted into a positive continuous signal of the same value.

When a continuous signal output is connected to a discrete signal input, the following rules apply. 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 is above the maximum discrete signal value, it is saturated to the maximum discrete signal value: 0xFFFFFFFF. All negative continuous signals are converted into zeros.

The undefined signals are converted into zeros unless the function block can process the undefined signal state, for example: *CAN Output Message* function block will output all ones for undefined signals.

3.2 Output Signal Sources

The controller output signal sources of all function blocks and source numbers are presented in the table below.

Table 5. Controller Signal Sources

Signal Source Number	Signal Name	Signal Type	Source Number
0	Not Connected	Undefined	0
1	Universal Input	Discrete or Continuous[1]	[1…4]
2	Thermocouple Temperature Input	Any[2]	[1…20]
3	Thermocouple Voltage Input	Any[2]	[1…20]
4	Thermocouple Input Raw Data	Any[2]	[1…20]
5	RTD Temperature Input	Any[2]	[1…2]
6	RTD Resistance Input	Any[2]	[1…2]
7	RTD Input Raw Data	Any[2]	[1…2]
8	CAN Input Signal	Any[2]	[1…3]
9	Math Function	Any[2]	[1…5]
10	Conditional Logic	Any[2]	[1…10]
11	Set-Reset Latch	Any[2]	[1…5]
12	Lookup Table	Any[2]	[1…10]
13	Programmable Logic	Any[2]	[1…3]
14	Global Continuous Constant Signal	Continuous	1
15	Global Discrete Constant Signal	Discrete	1
16	Supply Voltage	Continuous	1
17	Microcontroller Temperature	Continuous	1

[1] Depends on the *Input Parameter*.

[2] Depends on the *Signal Type* configuration parameter.

3.3 Universal Inputs

The *Universal Input* function block translates physical input signals into the internal function block output signal that can be used by other function blocks of the controller.

There are 4 independent *Universal Input* function blocks presenting their own universal physical inputs.

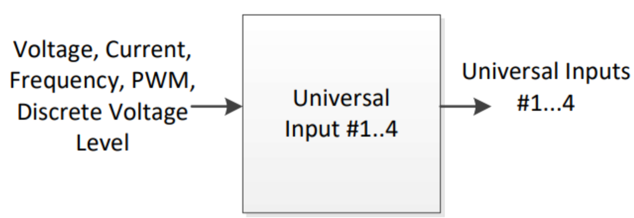


Figure 4. Universal Input Function Block

The internal function block output signal type and units of measurement are presented below.

Table 6. Universal Input Function Block Output Signal

Input Parameter	Type	Units
Voltage	Continuous	V
Current	Continuous	mA
Discrete Voltage Level	Discrete	{0,1}
Frequency	Continuous	Hz
PWM Duty Cycle	Continuous	%

Each *Universal Input* function block has the following configuration parameters.

Table 7. Universal Input Function Block Configuration Parameters

Parameter	Default Value	Range	Units	Description
Input Parameter	1 - Voltage	0 - Input Disabled, 1 - Voltage, 2 - Current, 3 - Discrete Voltage Level, 4 - Frequency, 5 - PWM Duty Cycle	–	Defines the input physical parameter that will be measured by the function block.
Voltage Range	0 - 0…5V	0 - 0…5 V, 1 - 0…10 V	V	Used in the "Voltage" mode
Current Range[1]	0 - 0…20 mA	0 - 0…20mA, 1 - 4…20 mA	mA	Used in the "Current" mode
Input Range Min	0	0…100	-	Depends on the Input Parameter. Used for diagnostic purposes
Input Range Max	5	0…100	-	Depends on the Input Parameter. Used for diagnostic purposes
Voltage LoZ Input	0 - No	0 - No, 1 - Yes	–	Activates a 10kOhm pull-down resistor to avoid ghost voltages in the "Voltage" mode. Warning: Measurement accuracy will be decreased!
Analog Input Filter	0 - Disabled	0 - Disabled, 1 - 50Hz Noise Rejection, 2 - 60Hz Noise Rejection, 3 - Both: 60Hz and 50Hz Noise Rejection	–	Noise Rejection in "Voltage", "Current" and “Resistance” modes
Pull-Up/Pull-Down Resistor	0 - Disabled	0 - Disabled, 1 - 10kOhm Pull-Up, 2 - 10kOhm Pull-Down	–	Used in "Discrete Voltage Level", "Frequency", and "PWM Duty Cycle" modes.
Input Polarity	0 - Active High	0 - Active High, 1 - Active Low	–	Used in "Discrete Voltage Level", "Frequency", and "PWM Duty Cycle" modes.
Discrete Input Debounce Time	50ms	0…1000	ms	Used in "Discrete Voltage Level" mode. If 0 - no debouncing.
Frequency Range	0 - 1Hz…10kHz	0 - 1Hz…10kHz,	Hz	A 16-bit counter is used. Used in "Frequency", and "PWM Duty Cycle" modes.

Parameter	Default Value	Range	Units	Description
Frequency/PWM Debounce Filter[3]	0 - Disabled	0 - Disabled, 1 - 142ns, 2 - 1.14us, 3 - 6.10us	–	Used in "Frequency", and "PWM Duty Cycle" modes.
Frequency/PWM Averaging	0 - No Averaging	0 - No Averaging, 1 - 3 Readings, 2 - 5 Readings, 3 - 10 Readings	–	Defines a moving average filer used in "Frequency", and "PWM Duty Cycle" modes.

[1] Input currents below 3mA are output as 0mA when 4…20 mA current range is set.

3.3.1 Voltage Measurements

The *Universal Inputs* can measure voltages in voltage ranges set by the *Voltage Range* configuration parameter.

To avoid an influence of ghost voltages, the *Voltage LoZ Input* configuration parameter can be activated. This will reduce the accuracy of voltage measurements due to the influence of the 10kOhm pull-down shunt resistor and should be used only after careful consideration of the shunt resistor influence on the measured circuit.

The user can set the *Analog Input Filter* configuration parameter to reduce noise in voltage and other analog signal measurements. The filter is designed to suppress noise from industrial offline voltages. Even when the analog input filter is disabled, the minimum signal filtering is performed by the function block. The parameters of the analog input filter are presented below.

Table 8. Universal Input Analog Input Filter Parameters

Analog Input Filter	Cut-off Frequency (at -3dB)	Settling Time (to 100% of Final Value)	Output Signal Update Rate
Disabled[1]	70Hz	10ms	1.67ms
50Hz Noise Rejection	12Hz	76.7ms	3.33ms
60Hz Noise Rejection	14Hz	63.3ms	3.33ms
Both: 60Hz and 50Hz Noise Rejection	2.3Hz	396.7ms	16.67ms

[1] Minimum filtering is still performed.

3.3.2 Current Measurements

There are two standard current ranges available for current measurements. When the current is below 3mA in the “4…20mA” current range, the output will be forced to zero to facilitate detection of an open circuit condition on the *Universal Input*.

The *Analog Input Filter* can be set to reduce the input noise.

3.3.3 Discrete Voltage Level

The *Universal Inputs* can accept discrete voltage levels. The user should specify the input polarity and define whether the pull-up/pull-down resistor is necessary on the input.

When the “10kOhm Pull-Up” is selected, the pull-up resistor is connected to the internal +14V power supply.

The input states are sampled every 1ms. If debouncing is required, it is set by the *Discrete Input Debounce Time* configuration parameter. If the *Discrete Input Debounce Time* is zero, the discrete voltage level input is not debounced.

3.3.4 Frequency and PWM

The frequency and PWM duty cycle measurements are performed by counting high-frequency internal clock pulses on every period of the input signal. The universal input channels have different internal organization due to limited hardware resources.

All universal inputs use 16-bit counters with the constant frequency range of 1...10kHz.

Table 9. Universal Input Function Block Counters

Function Block	Counter	Frequency Range	Counter Base	Shared Input	Frequency Range and Debounce Filter Setting
Universal Input #1	16-bit	1Hz…10kHz,	Dedicated	N/A	Same input
Universal Input #2			Dedicated	N/A	Same input
Universal Input #3			Dedicated	N/A	Same input
Universal Input #4			Dedicated	N/A	Same input

To measure frequency or PWM duty cycle, the user should first select the *Frequency Range* parameter and then define how the *Pull-Up/Pull-Down Resistor*, *Frequency/PWM Debounce Filter*, and the *Frequency/PWM Averaging* parameters should be set.

The *Input Polarity* defines the active edge of the input signal. The *Pull-Up/Pull-Down Resistor* can be used to pull the input to a no-signal state to avoid an undefined input condition when the signal source is disconnected. The *Input Polarity* and *Pull-Up/Pull-Down Resistor* are normally set the following way.

Table 10. Setting Pull-Up/Pull-Down Resistor for Selected Input Polarity. Universal Inputs

Input Polarity	Pull-Up/Pull-Down Resistor
Active High	“Disabled” or “10kOhm Pull-Down”
Active Low	“Disabled” or “10kOhm Pull-Up”

The frequency/PWM debounce filter is used to filter out parasitic spikes that can be present in a noisy input signal. It can be helpful to prevent the input from going into the Recovery state (see 3.3.4.1 Special Conditions) when, for example, mechanical switches are used to commutate the input signal.

The debounce filter should be used with caution since it can reduce the accuracy and resolution of frequency and PWM measurements if the debouncing time is not significantly less than the period of the input signal.

When a frequency or PWM signal presents a slowly changing parameter, setting an additional moving average filter using the *Frequency/PWM Averaging* configuration parameter can be helpful in smoothing the results of the input measurements.

3.3.4.1 Special Conditions

Frequencies below the Minimum Frequency value will be measured as zero and frequencies above the Maximum Frequency value will saturate at the Maximum Frequency value for the *Frequency Range*, see Table 11 and Table 12.

Table 11. Maximum, Minimum Frequencies and Maximum Recovery Time for Universal Inputs

Frequency Range	Counter	Minimum Frequency	Maximum Frequency	Maximum Recovery Time
1Hz…10kHz	16-Bit	0.9155Hz	12.5kHz	10.9ms

Frequencies above the Maximum Frequency value will switch the input to the Recovery state. The input will stay in the Recovery state until the upcoming counter saturation event when the frequency will be measured again. The input will leave the Recovery state if the measured frequency value is below the Maximum Frequency.

Table 12. Frequency and PWM Measurements for Universal Inputs. Special Conditions

Input Mode	Signal Frequency (F_s)			
	$F_s = 0$ Zero Frequency (DC)	$0 < F_s < F_{min}$ Below Minimum Frequency F_{min}	$F_{min} \leq F_s \leq F_{max}$ Working Frequency	$F_s > F_{max}$ Above Maximum Frequency F_{max}
Measured Frequency F_m	$F_m = 0$	$F_m = 0$	$F_m = F_s$	$F_m = F_{max}$ Recovery state
Measured PWM Duty Cycle D_m	$D_m = \{0, 100\}$	Undefined (not allowed)	$D_m = D_s$, D_s – signal duty cycle	$D_m = 0$ Recovery state

The time between two consequent counter saturation events defines the Maximum Recovery Time, see Table 11. This time is the maximum transient time when the measured frequency will stay equal to the Maximum Frequency value.

When the PWM signal is absent, the duty cycle is measured as 0 or 100% based on the voltage level on the input and the selected *Input Polarity*. The voltage level is sampled on the counter saturation events until the PWM signal is back on the input.

The transient time between the PWM signal duty cycle and the duty cycle of the DC level when the signal disappears can be up to the Maximum Recovery Time. During the transient time, the measured value will stay equal to the last measured value of the PWM signal duty cycle.

The PWM input signal with a frequency above zero but below the Minimum Frequency value is not allowed. The duty cycle will not be measured, instead, it will be jumping between 0% and 100% depending on the voltage level at the input on the counter saturation events.

When the PWM input signal frequency exceeds the Maximum Frequency value, the input goes into the Recovery state and the PWM duty cycle is measured as 0%. Like frequency measurements, the input will stay in the Recovery state for up to the Maximum Recovery Time before the duty cycle is measured again.

3.3.5 Diagnostics

The ECU has a diagnostic option for Measuring Out of Limits Faults: for low limit and high limit. The default values can be found in the Section 4.3.3. Disregarding the parameter set in the **Input Type**, the unit will use the value in **Input Range Min** and **Input Range Max** for the diagnostic high and low limits. The ranges for that parameter depend on the **Input Type** and can be found in the table below.

Table 13. Options for Input Range Min and Max Ranges

Input Type	Range
0	Input Disabled
1, Voltage	0…10
2, Current	0…24
3, Digital	0…1
4, Frequency	0…10000
5, PWM	0…100

3.4 Thermocouple Input Function Block

The ECU has 20 Thermocouple inputs, which can be configured to read the input data and react in a variety of different ways.

The first parameter, **Thermocouple Type**, is used to configure what type of Thermocouple the input is.

Table 14: Thermocouple Types

Value	Meaning
0	Input Disabled
1	B Type
2	E Type
3	J Type
4	K Type
5	N Type
6	R Type
7	S Type
8	T Type

3.4.1 Thermocouple Input Cold Junction Compensation

A high accuracy digital temperature sensor is placed next to thermocouple connectors to provide cold junction compensation. Cold Junction Compensation can be enabled or disabled for the Thermocouple Input using the **Cold Junction Enabled** setpoint. Regardless of this parameter being enabled, output data from the cold junction will still be available as a control source for outputs or other logic blocks.

3.4.2 Thermocouple Input Diagnostic Parameters

The **Generate Diagnostic Messages**, and **SPN for Diagnostics** setpoints are directly linked to all Diagnostic blocks related to the Thermocouple. This serves as a way to unilaterally change all these parameters at once for the affected diagnostics. Also, the ECU is able to detect the Open Circuit Fault. See Section 3.14 for more details.

3.4.3 Thermocouple Input Warning and Shutdown

The temperature levels at which the Shutdown and Warning messages are triggered are configurable. The **High Shutdown Temperature** and **Low Shutdown Temperature** parameters are used to set the upper and lower bound for when a Shutdown Diagnostic message is sent. The **High Warning Temperature** and **Low Warning Temperature** parameters are used to set the upper and lower bound for when a Warning Diagnostic message is sent.

The **Shutdown Delay** and **Warning Delay** parameters are used to set the delay between the trigger and actual output of a diagnostic message. These parameters are directly linked to the **Delay Before Sending DM1** setpoint for their respective diagnostic block.

3.5 RTD Input Function Block

The RTD Input Function Block reads analog signal from two RTD sensor inputs and processes them into the temperature data.

The user can specify the type of sensor connection (2-wire or 3-wire) and approximation parameters for each individual sensor connected to the unit using setpoint parameters. Axiomatic EA software is used for this purpose. The setpoint parameters are kept in a non-volatile memory of the RTD unit and are automatically loaded on power-up.

Although the diagnostics check the temperature data, the ECU is capable of providing the output data in Ohms as well as a raw hexadecimal data that has been read from a RTD chip.

The AX180800 supports the PT100 sensor type in both 2- and 3-wire configuration. The supported sensor connections are shown in the table below.

Table 15 The Types of RTD Connections

Value	Description
0	RTD Disabled
1	Two Wires
2	**Three Wires**

3.5.1 RTD Coefficients

The customer can set the specific coefficients within ranges. By default, the values for Callendar - Van Dusen Coefficient A, B, and C are standard (IEC 0.00385), and can be seen on the table below.

Table 16 The Default Value of Coefficients

Name	Default Value	Range	Units	Description
Callendar – Van Dusen coefficient A	3.908300	-100…100	–	Callendar – Van Dusen coefficient A for the selected standard coefficient set. Editable, if the coefficient set is “User Defined”.
Callendar – Van Dusen coefficient B	-5.77500	-100…100	–	Callendar – Van Dusen coefficient B for the selected standard coefficient set. Editable, if the coefficient set is “User Defined”.
Callendar – Van Dusen coefficient C	-4.183010	-100…100	–	Callendar – Van Dusen coefficient C for the selected standard coefficient set. Editable, if the coefficient set is “User Defined”.

If these values need to be changed, the user can change them via EA or Modbus.

3.5.2 Warning and Shutdown Limits

The AX180800 has High and Low Limits for all types of output: °C, mV, and Raw Data. Also, there is High and Low Shutdown Limit for Shutdown Temperature Fault Diagnostics. Even though only temperature reading is used for the diagnostics, the customer can use the voltage and raw data reading as control sources (See Table 5). The user can set the value for limits via EA or Modbus. The default setpoints can be found in Section 4.3.2.

3.6 Relay Output Function Block

The following sub-sections will explain in more detail the functionalities and available setpoints/parameters.

3.6.1 Relay Output Functionality

The relay output has 2 states: Normally Open and Normally Closed. It has 3 pins associated with it: Normally Closed (NC), Normally Open (NO), and Common (C). The **“Relay Output Type”** parameter allows for flexibility in the response of the output. Table 17 shows the options available for this parameter.

Table 17. Relay Output Types

Value	Meaning
0	*Relay Disabled*
1	*Normal Logic*
1	*Inverse Logic*

2	*Latched Logic*
3	*Inverse Latched Logic*
4	*Toggle Logic*

By default, *'Normal Logic'* response is used for the relay outputs. In *'Normal Logic'* response, the Common pin is connected to the Normally Closed pin if the source of the respective relay output is triggered ON, the Common pin is connected to the Normally Open pin.

In the case of *'Inverse Logic'* response, the Common pin is connected to the Normally Open pin.
when the source of the respective relay output is triggered ON. When the source of the respective relay output is triggered OFF, the Common pin is connected to the Normally Closed pin.

In the case of *'Latched Logic'* response, the Common pin is toggled between Normally Closed and
Normally Open pins every time the source of the respective relay output goes from OFF to ON. The *'Inverse Latched Logic'* response will respond the opposite way.

The *'Toggle Logic'* lets the relay output toggle between Normally closed and Normally Open pins for a configured frequency. The time for switching from one state to the other state results in the **"Relay Toggle Rate"** which is in milliseconds and by default 500ms.

3.6.2 Relay Output Control / Enable Sources / Override Source

The relay output can be configured to be commanded and/or enabled by the control sources listed in Table 5. This table also displays the number associated to the control sources which can be selected. The default control source is highlighted while the default Enable Source and Override Source is configured to *'Control Not Used'.*

The selected control source in the **"Relay Control Source"** parameter is the main commanding source of the relay output based on "**Relay Output Type"** parameter. A delay can be set for both output states when **"Relay Enable Response Delay"** is set to be *'TRUE'.* In case the output state should turn low after a certain amount of time, the parameter **"Relay Delay OFF Time"** can be set. Whereas the **"Relay Delay ON Time"** can be configured to set a delay before switching from the OFF-state to ON-state. Both delays are configurable in milliseconds.

3.6.3 Relay Output Enable

The **"Relay Enable Source"** will determine whether or not the relay output will be commanded by the **"Relay Control Source"**. There are six different **"Relay Enable Response"** in which the enable signal can be used. These responses are listed in Table 18.

Table 18: Relay Enable Response

Value	Meaning
0	*Enable When ON*
1	*Enable When OFF*
2	*Disable When ON*

3	*Disable When OFF*
4	*Enable When ON Else Keep State*
5	*Enable When OFF Else Keep State*

When the **"Relay Enable Response"** is set to *'Enable When ON'* or *'Disable When OFF'*, the relay output will be commanded according to the combined signal of the **"Relay Control Source"** and **"Relay Control Number"** only when the signal of the **"Relay Enable Source"** and **"Relay Enable Number"** is ON. Otherwise, the relay output is commanded to the OFF state.

Similarly, when the **"Relay Enable Response"** is set to *'Enable When OFF'* or *'Disable When ON'*, the relay output will be commanded according to the **"Relay Control Source"** and **"Relay Control Number"** only when the signal of the **"Relay Enable Source"** and **"Relay Enable Number"** is OFF. Otherwise, the relay output is commanded to the OFF state.

In case the **"Relay Enable Response"** is *'Enable When ON Else Keep State'*, the relay output will be commanded according to the signal of the **"Relay Control Source"** and **"Relay Control Number"** only when the signal of the **"Relay Enable Source"** and **"Relay Enable Number"** is ON. If the Enable Signal is OFF, the relay output will keep the previous state.

Likewise, when the **"Relay Enable Response"** is configured to *'Enable When OFF Else Keep State'*, the relay output will be commanded according to the **"Relay Control Source"** and **"Relay Control Number"** only when the combined signal of **"Relay Enable Source"** and **"Relay Enable Number"** is OFF. Otherwise, the relay output holds the previous state.

3.6.4 Relay Output Override

The **"Relay Override Source"** will determine whether or not the relay output will be commanded by the **"Relay Control Source"**. This Source has a higher priority than the Enable Source.

There are two different **"Relay Override Response"** in which the Override signal can be used. These responses are listed in *Table 19*.

Table 19: Relay Override Response Options

Value	Meaning
0	*Override When OFF*
1	*Override When ON*

When the "Relay Override Response" is configured to 'Override When ON', the relay output will be commanded according to the signal of the "Relay Control Source" and "Relay Control Number" by the "Relay Override State" only when the override signal is ON. If the "Relay Override Response" is set to 'Override When OFF', the relay output will be commanded only according to the signal of the Control Source/Number by the "Relay Override State" only when the override signal is OFF.

Table *20* shows the two possible states for the "**Relay Override State**".

In the case of *'Override State OFF'*, the relay output switches to Normally Open. If *'Override State ON'* is configured, the relay output changes to Normally closed.

Table 20. Relay Override State Options

Value	Meaning
0	*Override State OFF*
1	*Override State ON*

3.6.5 Unlatch Source

This Source can only be configured if the **"Relay Output Type"** is set to *'Latched Logic* or *'Inverse Latched Logic*' and it can be enabled/disabled by the parameter **"Relay Enable Unlatch Source"**. If the signal of the **"Relay Unlatch Source"** is ON, it turns the output OFF when the **"Relay Output Type"** is set to *'Latched Logic*'. If the Unlatch Source state turns OFF afterwards, the output state stays OFF independent of the output state before. The reverse behavior is applied to the *Inverse Latched Logic*.

3.7 Math Function Block

There are five mathematical function blocks that allow the user to define basic algorithms. A math function block can take up to six input signals. Each input is then scaled according to the associated limit and scaling setpoints.
Inputs are converted into percentage value based on the "**Input X Minimum**" and "**Input X Maximum**" values selected. For additional control the user can also adjust the "**Input X Decimal Digits**" setpoint to increase the resolution of the input data and the min and max values.
A mathematical function block includes three selectable functions, in which each implements equation $\mathrm{A\ operator\ B}$, where A and B are function inputs and operator is function selected with a setpoint "**Math Function X**". Setpoint options are presented in Table *22. Math function X Operator Options*

. The functions are connected, 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.

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

Table 21. Math function X Operator Options

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 22. Math function X Operator Options

For logic operations (6, 7, and 8) scaled input greater than or equal to 1 is treated as TRUE. For logic operations (0 to 8), the result of the function will always be 0 (FALSE) of 1 (TRUE). For the arithmetic functions (9 to 14), it is recommended to scale the data such that the resulting operation will not exceed full scale (0 to 100%) and saturate the output result. When dividing, a zero divider will always result in a 100% output value for the associated function.
Lastly the resulting mathematical calculation, presented as a percentage value, can be scaled into the appropriate physical units using the "**Math Output Minimum Range**" and "**Math Output Maximum Range**" setpoints. These values are also used as the limits when the Math Function is selected as the input source for another function block.

3.8 Conditional Block

The Conditional Block compares up to four different input sources with different logical or relational operators. The result of each block can therefore only be true (1) or false (0). Figure 5 demonstrates the connections between all parameters.

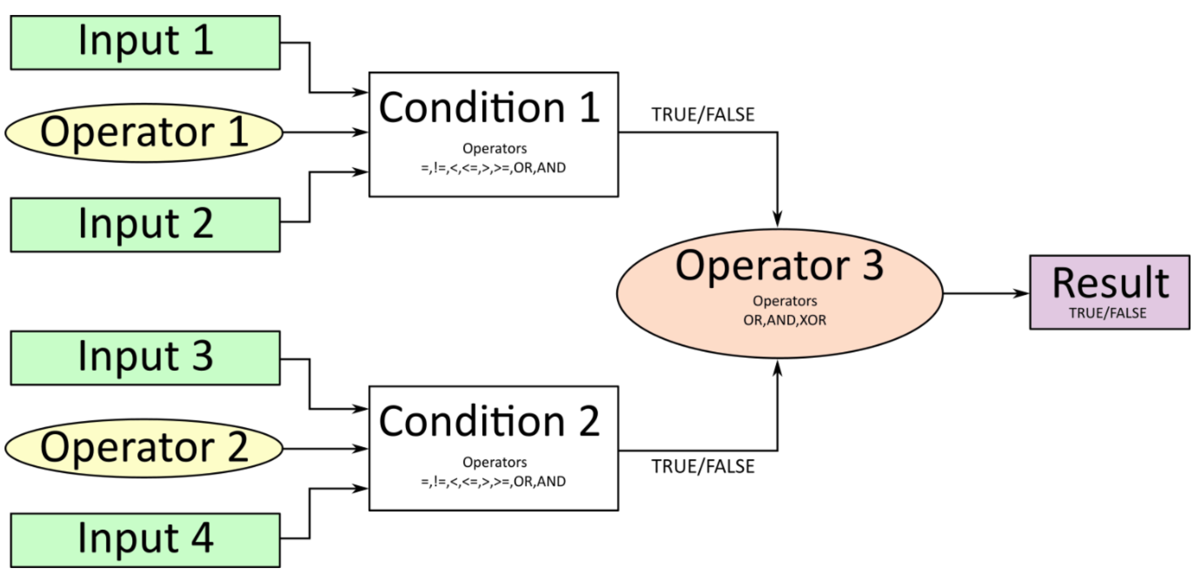


Figure 5: Conditional Block Diagram

Each Conditional Block offers two conditions. Both compare two inputs, which can hold a logical value or an integer value. The output of the conditions can only be true or false and will be compared by Operator 3 with a logical operator. This comparison is the result of the Conditional Block and can control any output source.

value of each source will then be compared to each other with an operator of Table 23. If no source is selected, the output value of an Input will be zero.

Table 23. Input Operator Options

Value	Meaning
0	==, True when Argument 1 is equal to Argument 2
1	!=, True when Argument 1 is not equal to Argument 2
2	>, True when Argument 1 is greater than Argument 2
3	>=, True when Argument 1 is greater than Argument 2
4	<, True when Argument 1 is less than Argument 2
5	<=, True when Argument 1 is less than or equal Argument 2
6	OR, True when Argument 1 or Argument 2 is True
7	AND, True when Argument 1 and Argument 2 are True

Operator 1 and Operator 2 are configured to OR by default. The table above cannot be used for comparing the conditions because they can only be compared with logical operators, which are listed in Table 24.

Table 24. Condition Operator Options

Value	Meaning
0	OR, True when Argument 1 or Argument 2 is True
1	AND, True when Argument 1 and Argument 2 are True
2	XOR, True when Argument 1 is not equal to Argument 2

If only one condition is used, it is to make sure that Operator 3 is set to **OR** so that the result is based solely on the condition which has been chosen.

3.9 Set / Reset Latch Function Block

Set-Reset Block consists of only 2 control sources: **Reset Source** and **Set Source**. The purpose of these blocks is to simulate a modified latching function in which the 'Reset Signal' has more precedence. The 'latching' function works as per the Table 25 below.

Table 25. Set-Reset Function block operation.

'Set Signal'	'Reset Signal'	'Set-Reset Block Output' **(Initial State: *OFF*)**
OFF	*OFF*	*Latched State*
OFF	*ON*	*OFF*
ON	*OFF*	*ON*
ON	*ON*	*OFF*

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

As seen in Table 25 above, the 'Reset Signal' has more precedence over the 'Set Signal' - if the state of 'Reset Signal' is *ON*, the state of 'Set-Reset Block Output' will be *OFF*. To create an *ON* state in 'Set-Reset Block Output' the state of 'Reset Signal' must be *OFF* while the state of 'Set Signal' is *ON*. In this case, the state of 'Set-Reset Block Output' will remain *ON* even if 'Set Signal' turns *OFF* as long as 'Reset Signal' remains *OFF.* As soon as the 'Reset

Signal' turns *ON* the 'Set-Reset Block Output' will turn *OFF* regardless of the state of 'Set Signal'.

3.10 Lookup Table Function Block

Lookup Tables are used to give output response up to 10 slopes per input. 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 described in Section 3.11. Lookup tables have two differing modes defined by "**X-Axis Type**" setpoint, given in Table 26. Option '0 – Data Response' is the normal mode where block input signal is selected with the "**X-Axis Source**" and "**X-Axis Number**" setpoints and X values present directly input signal values. With option '1 – Time Response' the input signal is time and X values present time in milliseconds. And selected input signal is used as digital enable.

Table 26. X-Axis Type Options

0	*Data Response*
1	*Time Response*

The slopes are defined with (x, y) points and associated point response. X value presents input signal value and Y value corresponding Lookup Table output value. "PointN – Response" setpoint defines type of the slope from preceding point to the point in question. Response options are given in Table 27. 'Ramp To' gives a linearized slope between points, whereas 'Jump to' gives a point to point response, where any input value between X_{N-1} and X_N will result Lookup Table output being Y_N. "Point0 – Response" is always 'Jump To' and cannot be edited. Choosing 'Ignored' response causes associated point and all the following points to be ignored.

Table 27. PointN – Response Options

0	*Ignore*
1	*Ramp To*
2	*Jump To*

The X values are limited by minimum and maximum range of the selected input source if the source is a Math Function Block. For the fore mentioned sources X-Axis data will be redefined when ranges are changed, therefore inputs should be adjusted before changing X-Axis values. For other sources Xmin and Xmax are -100000 and 1000000. The X-Axis is constraint to be in rising order, thus value of the next index is greater than or equal to preceding one. Therefore, when adjusting the X-Axis data, it is recommended that X_{10} is changed first, then lower indexes in descending order.

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

The Y-Axis has no constraints on the data it presents, thus inverse, decreasing, increasing or other response can be easily established. The Smallest of the Y-Axis values is used as Lookup Table output min and the largest of the Y-Axis values is used as Lookup Table output max (i.e. used as Xmin and Xmax values in linear calculation.). Ignored points are not considered for min and max values.

3.11 Programmable Logic Function Block

The Programmable Logic Function Block is a powerful tool. Programmable Logic can be linked to up to three Lookup Tables, any of which would be selected only under given conditions. Thus, the output of a Programmable Logic at any given time will be the output of the Lookup Table selected by the defined logic. 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.

To enable any one of the Programmable Logic blocks, the "**Logic Enabled**" setpoint must be set to '*True'*. By default, all Logic blocks are disabled.

The three associated tables are selected by setting "**Table Number X**" setpoint to desired Lookup Table number, for example selecting *1* would set Lookup Table 1 as TableX.

For each TableX there are three conditions that define the logic to select the associated Lookup Table as Logic output. Each condition implements function $Argument1\ Operator\ Argument2$ where Operator is logical operator defined by setpoint "**Table X – Condition Y Operator**". Setpoint options are listed in Table 28. Condition arguments are selected with "**Table X – Condition Y Argument Z Source**" and "**Table X – Condition Y Argument Z Number**" setpoints. If '*0 – Control not Used*' option is selected as "**Table x – Condition Y Argument Z Source"** the argument is interpreted as 0.

Table 28. Table X – Condition Y Operator Options

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

The three conditions are evaluated and if the result satisfies logical operation defined with "**Logical Operator X**" setpoint, given in Table 29, the associated Lookup Table is selected as output of the Logical block. Option '*0 – Default Table*' selects associated Lookup Table in all conditions.

Table 29. Table X – Conditions Logical Operator Options

0	*Default Table (Table1)*
1	*Cnd1 And Cnd2 And Cnd3*
2	*Cnd1 Or Cnd2 Or Cnd3*
3	*(Cnd1 And Cnd2) Or Cnd3*
4	*(Cnd1 Or Cnd2) And Cnd3*

The three logical operations are evaluated in order and the first to satisfy gets selected, thus if Table1 logical operation is satisfied, the Lookup Table associated with Table1 gets selected regardless of two other logical operations. In addition, if none of the logical operations is satisfied the Lookup Table associated with Table1 gets selected.

3.12 Global Parameters

The *Global Parameters* functional block gives the user access to a set of global constants, unit supply voltage and the microcontroller internal temperature.

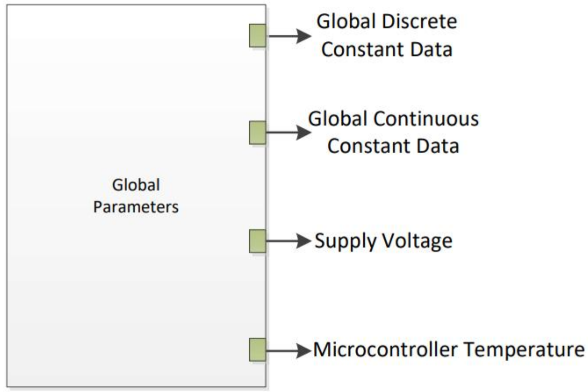


Figure 6. Global Parameters Function Block

The function block provides two configurable global constant signals. It also provides the "Supply Voltage" signal presenting the controller supply voltage in [V] and the "Microcontroller Temperature" signal presenting the internal microcontroller temperature in [°C].

Please note, that the "Supply Voltage" signal does not present the voltage on the controller power supply connector pins. It shows an internal voltage measured after the EMI filter, reverse polarity, and transient protection circuit. The reading accuracy is within the range of 1V.

The *Global Parameters* function block has the following configuration parameters.

Table 30. Global Parameters Function Block Configuration Parameters

Name	Default Value	Range	Units	Description
Global Continuous Constant Signal	0	Any value	–	Signal value of the *Global Continuous Constant Signal.*
Global Discrete Constant Signal	0	0… 4294967295 (0xFFFFFFFF)	–	Signal value of the *Global Discrete Constant Signal.*

3.13 Miscellaneous

The Miscellaneous function block contains various parameters that affect the general diagnostic performance of the ECU.

The **Undervoltage Threshold**, **Overvoltage Threshold**, and **Shutdown Temperature** setpoints are used to set the limits for when their respective diagnostic messages are triggered.

Lastly, the **CAN Diagnostic Setting1** and **CAN Diagnostic Setting2** parameters are used to control all diagnostics with one general setting for CAN Interface 1 and CAN Interface 2 respectively. This can be used to disable diagnostics entirely, only transmit messages without a blank SPN, or transmit diagnostic messages normally.

3.14 Diagnostics

The Diagnostic function block includes 120 faults, each representing a diagnostic message that the ECU can produce. Each Universal Input has a Voltage Out of Range Low and

Voltage Out of Range High Faut. Each RTD has Temperature Out of Range Low and Temperature Out of Range High, High and Low Shutdown Faults. Each Thermocouple Input has Temperature Out of Range Low and Temperature Out of Range High, High and Low Shutdown, and Open Circuit Faults. The remaining faults cover VPS Overvoltage and Undervoltage, Overtemperature, and other faults.

If and only if the **Event Generates a DTC in DM1** parameter 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 4-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 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 Controller will send the DM1 message every one second, regardless of whether there are any active faults, as recommended by the standard. While there are no active DTCs, the Controller will send the “No Active Faults” message. If a previously active DTC becomes inactive, 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 on 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 broadcast 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.

The Diagnostic function block has a setpoint **Event Cleared Only by DM11**. By default, this is 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 change, 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. In those cases, the FMI in the setpoint reflects that of the low-end condition, and the FMI used by the high fault will be determined per Table 31. If the FMI is changed, the OC of the associate error log is automatically reset to zero.

Table 31. Low Fault FMI versus High Fault FMI

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

If the FMI used is anything other than one of those in Table 31, then both the low and the high faults will be assigned the same FMI. This condition should be avoided, as the log will still use different OC for the two types of faults, even though they will be reported the same in the DTC.

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 the Diagnostic function block. If the fault has remained present during the delay time, then the controller will set the DTC to active, and it 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.

3.15 J1939 Network

The *J1939 Network* function block defines the global J1939 CAN bus settings. It does not have signal inputs and outputs.

Figure 7. J1939 Network Function Block

Configuration parameters of the *J1939 Network* function block are presented below. They contain *ECU Network* and *CAN Network Parameters*.

Table 32. J1939 Network Function Block Configuration Parameters

Name	Default Value	Range	Units	Description
ECU Instance Number	0 - Instance #1	0…7	–	ECU Instance field of the J1939 ECU Name.
ECU Address	128	0…253	–	J1939 ECU address.
Baud Rate[1]	250	{250, 500, 667, 1000}	kbit/s	Current baud rate on the CAN network.
Automatic Baud Rate Detection	1 - Yes	0 - No, 1 - Yes	–	Set to “No” once ECU is permanently installed on the CAN network.
Slew Rate	0 - Low	0 - Low, 1 - High	–	Slew rate control of the CAN transceiver.

[1] Read-only parameter.

3.15.1 ECU Network Parameters

The user can change the *ECU Instance Number* and *ECU Address* to adjust the unit on the CAN network.

Changing the *ECU Instance Number* is necessary to accommodate multiple inclinometers on the same CAN network. The list of available ECU instances is shown in the *ECU Instance Number Setup* dialog window in EA. The user should select the required ECU instance number and then press OK or double-click the selected instance number.

The *ECU Address* is automatically adjusted as the result of an address arbitration process on the J1939 CAN network. It can also be changed by a commanded address message. The user can also manually change the ECU address using the *ECU Address* configuration parameter.

The user selects the new ECU address from the list of available ECU addresses in the *ECU Address Setup* dialog window in EA. After the required ECU address is selected, the user should press OK button or double-click the selected address.

3.15.2 CAN Network Parameters

The *Baud Rate* read-only configuration parameter shows the current baud rate on the CAN network.

The *Automatic Baud Rate Detection* parameter defines whether the ECU will try to detect the CAN baud rate in case of communication errors. The baud rate is detected from the list of supported CAN baud rates.

To avoid an arbitrary selection of the CAN baud rate by ECUs involved in the automatic baud rate detection process, it is necessary to disable the automatic baud rate detection in ECUs that are already permanently installed on the CAN network.

The *Slew Rate* configuration parameter defines the slew rate of the CAN transceiver the following way:

Table 33. Slew Rates

Slew Rate Value	Transceiver Slew Rate	Note
Fast	~40 V/µs	Available for all baud rates.
Slow	~6 V/ µs	Only available for 250kbit/s baud rate.

The user can select the *Slew Rate* only when the inclinometer operates at 250 kbit/s baud rate. For baud rates higher than 250 kbit/s, the *Slew Rate* is always set to “Fast” independently of the *Slew Rate* configuration parameter.

The “Slow” slew rate is preferable at 250 kbit/s baud rate in the majority of applications due to the reduced EMI of the CAN transceiver. The “Fast” slew rate, in this case, is used when the distance between CAN nodes substantially exceeds 40 m – the maximum value defined by the J1939/11(15) standard.

3.16 Constant Data

The Constant Data Block contains four configurable constant data setpoints which can be used as a control source for other functions. While they are available as a control source to all functions, it is recommended not to use constant data as a control source for the Set-Reset Latch Block.

3.17 Ethernet

The *Ethernet* function block defines the Modbus TCP interface settings. It does not have signal inputs and outputs.

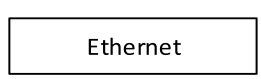


Figure 8. Ethernet Function Block

Configuration parameters of the *Ethernet* function block are presented below.

Table 34. Ethernet Function Block Configuration Parameters

Name	Default Value	Range	Units	Description
MAC Address	Set at the factory	Any valid MAC address	–	Ethernet MAC Address. Set at the factory. Read-only parameter.
IP Address	192.168.1.34	Any IP address	–	The device IP address
Subnet Mask	255.255.255.0	Any IP address	–	The device subnet mask
Gateway	192.168.1.1	Any IP address	–	The device default gateway
Modbus Port	502	Any port value except the Discovery Port (35100)	–	The Modbus listening port
Modbus Timeout	1000	1…10000	ms	The Modbus communication timeout. Not used in the current firmware.

Any updates to the function block configuration parameters will require a manual reset of the controller to apply the new Ethernet settings.

3.18 Modbus Receive Function Block

The Modbus Receive function block allows ECU to receive up to 16 bit of Modbus incoming data from a chosen register.
To enable the block, ***"Signal Enabled"*** *setpoint should be set to 1, True.* The register address can be chosen from a predefined range of addresses and is configured by setting the ***"Register Address"*** setpoint. The range is shown in a Modbus Address Map. The ***"Data Size in Bits"*** and ***"Data Position Bit"*** setpoints help to process the register data, so the unused bits will be stripped.
Please note: the incoming data cannot exceed 16 bits and is limited to one register. Thus, if ***"Data Size in Bits"*** and ***"Data Position Bit"*** configuration exceeds the register limits, everything extra will be stripped.
Also, the Modbus Receive function block has a resolution and offset that are applied at last. Overall, the output data will be processed in the following way:

$$Out_{PV} = ((In_{FV} \& Mask) \gg Position) * Resolution + \ Offset$$

where In_{PV} – Input register data,
Mask – a bit mask that is calculated based on the data size and the position set,
Position – a value set in ***"Data Position Bit"***,
Resolution and Offset – ***"Resolution"*** and ***"Offset"*** setpoint values

If the data is planned to be received within a certain timeframe, the Modbus Receive function block has an auto resetting feature. The controller can be configured to set the output data to 'Not Available' if the last data wasn't received within the "***Autoreset Time"*** in ms. If the "***Autoreset Time"*** setpoint is set to 0 ms, this feature will be disabled.

3.19 CAN Input Signals

There are three *CAN Input Signal* function blocks available to the user. Each function block represents one CAN input signal that can be received from the CAN bus. The function block has one signal output.

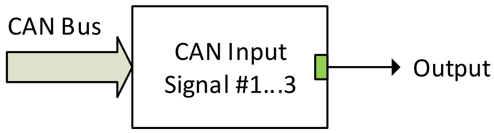


Figure 9. CAN Input Signal Function Block

The *CAN Input Signal* function block reads single-frame application-specific CAN messages and extracts CAN signal data presented in a user-defined data format. Different *CAN Input Signal* function blocks can read and process the same CAN message to extract different CAN signal data.

The CAN messages transmitted by the unit itself are also processed by *CAN Input Signal* function blocks. The only difference in processing of the internal messages is that they are not sampled from the CAN bus and therefore their processing does not depend on the state of the CAN bus.

Configuration parameters of the *CAN Input Signal* function block are presented below.

Table 35. CAN Input Signal Function Block Configuration Parameters

Name	Default Value	Range	Units	Description
CAN Interface	1, CAN 1	0 - 3	–	Defines the CAN interface used to send messages (CAN 1, CAN 2 or Both)
Signal Type	0 - Undefined	0 - Undefined, 1 - Discrete, 2 - Continuous	–	CAN input signal type
PGN	65535	Any J1939 PGN value[1]	–	Signal message PGN value
PGN From Selected Address	0 - No	0 - No, 1 - Yes	–	Only CAN messages from the selected address will be accepted, if “Yes”
Selected Address	0	0…253	–	Address of the ECU transmitting CAN messages if *PGN From Selected Address* is set to “Yes”
Data Position Byte	1	1…8	–	Start byte of the CAN input signal in the CAN message data frame
Data Position Bit	1	1…8	–	Start bit of the CAN input signal in the *Data Position Byte*
Size	1	1…32	bit	CAN input signal size.
Resolution	1	Any value	signal units / bit	CAN input signal resolution for continuous input signals
Offset	0	Any value	signal units	CAN input signal offset for continuous input signals

Name	Default Value	Range	Units	Description
Autoreset Time	500	0…10000	ms	Function block signal output auto-reset time. If *Autoreset Time* is 0, the auto-reset is disabled.

[1]*Proprietary A PGN (61184) is excluded. It is taken by Axiomatic Simple Proprietary Protocol and therefore cannot be used in function blocks.*

The CAN input signal position is defined within the CAN message data frame by the *Data Position Byte* and *Data Position Bit* configuration parameters the same way as in the J1939 standard. The start and stop bits of the CAN signal in the 64-bit CAN message data frame are calculated using the formulas:

$$StartBit = (DataPositionByte - 1) \cdot 8 + (DataPositionBit - 1), \quad (1)$$

$$StopBit = StartBit + Size - 1, where: StartBit, StopBit \in [0 \dots 63].$$

Resolution and *Offset* configuration parameters are set for continuous CAN input signals. They are not used for discrete CAN signals.

The following rules apply when converting the CAN signal data to the function block output signal:

- Undefined CAN signals with all bits set to 1 are ignored.
- Discrete CAN signals can take any value except the one reserved for the undefined signal (all bits set to 1).
- Continuous CAN signals can take only values from the range reserved for continuous signals in the J1939 standard. If the CAN signal code is outside of this range presenting a special condition or an error, the signal is ignored.

When the *Autoreset Time* is not equal to 0, the function block will auto-reset the output signal to the undefined state if the output signal has not been updated within the auto-reset time frame by the new CAN message data.

3.20 CAN Output Messages

There are five *CAN Output Message* function blocks available to the user. Each function block presents one single frame CAN output message that can be sent on the CAN bus. The message can contain up to ten CAN output signals. Each CAN output signal is presented by its own signal input in the function block.

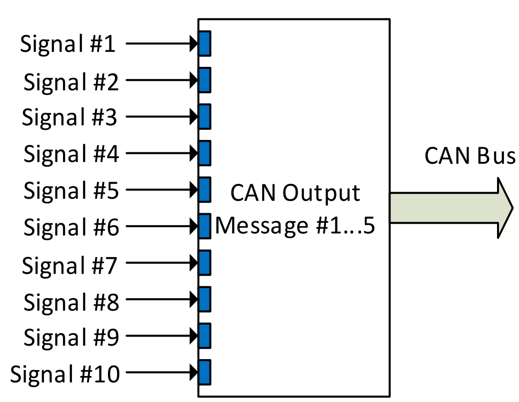


Figure 10. CAN Output Message Function Block

Configuration parameters of the *CAN Output Message* function block are presented below.

Table 36. CAN Output Message Function Block Configuration Parameters

Name	Default Value	Range	Units	Description
CAN Interface	1, CAN 1	0 - 3	–	Defines the CAN interface used to send messages (CAN 1, CAN 2 or Both)
PGN	65535	Any J1939 PGN value[1]	–	CAN message PGN.
Transmission Enable	0 - No	0 - No, 1 - Yes	–	Enables the CAN output message transmission.
Transmission Rate	10	0…10000	ms	CAN output message transmission rate. If *0* – transmission is upon request.
Destination Address	255	0…255	–	Destination address of the PDU1 PGN messages.
Length	8	0…8	byte	CAN message data frame length.
Priority	6	0…7	–	CAN message priority.
Signal #1 Type	0 - Undefined	0 - Undefined, 1 - Discrete, 2 - Continuous	–	Type of the 1-st CAN output signal.
Signal #1 Source	0 - Not Connected	Any signal output of any function block or “Not Connected". See Signal Source table.	–	Input signal source of the 1-st CAN output signal.
Signal #1 Byte Position	1	0…8	–	Byte position of the 1-st CAN output signal.
Signal #1 Bit Position	1	0…8	–	Bit position of the 1-st CAN output signal.
Signal #1 Size	1	0…32	bit	Size of the 1-st CAN output signal.
Signal #1 Resolution	1	Any value	signal units / bit	Resolution of the 1-st CAN continuous output signal.
Signal #1Offset	0	Any value	signal units	Offset of the 1-st CAN continuous output signal.
Signal #2 Type	0 - Undefined	0 - Undefined, 1 - Discrete, 2 - Continuous	–	Type of the 2-nd CAN output signal.
Signal #2 Source	0 - Not Connected	Any signal output of any function block or “Not Connected". See Signal Source table.	–	Input signal source of the 2-nd CAN output signal.
Signal #2 Byte Position	1	1…8	–	Byte position of the 2-nd CAN output signal.
Signal #2 Bit Position	1	1…8	–	Bit position of the 2-nd CAN output signal.
Signal #2 Size	1	1…32	bit	Size of the 2-nd CAN output signal.

Name	Default Value	Range	Units	Description
Signal #2 Resolution	0	Any value	signal units / bit	Resolution of the 2-nd CAN continuous output signal.
Signal #2 Offset	1	Any value	signal units	Offset of the 2-nd CAN continuous output signal.
…	…	…	…	…
Signal #10 Type	0 - Undefined	0 - Undefined, 1 - Discrete, 2 - Continuous	–	Type of the 10-th CAN output signal.
Signal #10 Source	0 - Not Connected	Any signal output of any function block or “Not Connected". See Signal Source table.	–	Input signal source of the 10-th CAN output signal.
Signal #10 Byte Position	1	1…8	–	Byte position of the 10-th CAN output signal.
Signal #10 Bit Position	1	1…8	–	Bit position of the 10-th CAN output signal.
Signal #10 Size	1	1…32	bit	Size of the 10-th CAN output signal.
Signal #10 Resolution	1	Any value	signal units / bit	Resolution of the 10-th CAN continuous output signal.
Signal #10 Offset	0	Any value	signal units	Offset of the 10-th CAN continuous output signal.

[1] *Proprietary A PGN (61184) is excluded. It is taken by Axiomatic Simple Proprietary Protocol and therefore cannot be used in function blocks.*

Configuration parameters: *Signal #1…10 Byte Position* and *Signal #1…10 Bit Position*, together with the *Signal #1…10 Size* have the same meaning as in the *CAN Input Signal* function block. The user should be careful not to overlap the output signals.

The following rules apply when converting function block output signal data to the CAN signal code:

- Undefined signals are presented in the CAN signal code with all bits set to 1.
- Discrete signals are directly assigned to the CAN signal code without any conversion.
- Continuous signals are converted to the CAN signal code based on the *Signal #1…10 Resolution* and *Signal #1…10 Offset* configuration parameters. They are saturated to the CAN continuous signal code boundaries defined in the J1939 standard when they go out of range.

4 CONTROLLER CONFIGURATION

The controller can be configured in two independent ways: through the Modbus or CAN interface.

4.1 Modbus Configuration

The controller can be configured through the Modbus TCP interface using any third-party software tools.

The configuration parameters are grouped by the function block.

The controller checks configuration parameters for validity before accepting them and writing in a non-volatile memory. If a configuration parameter is invalid, the Modbus writing function will succeed, but the configuration parameter will not be written.

The controller will reset all relevant function blocks after each change of the configuration parameters. The exception is the *Ethernet* function block that maintains the Ethernet connection. The user will need to perform a manual reset by cycling the controller power to start using the new Ethernet settings.

Any changes in CAN function blocks through Modbus will restart CAN communication of the controller.

4.2 CAN Configuration

The controller supports the J1939 memory access protocol for configuring the unit through the CAN interface. Axiomatic's PC-based Electronic Assistant (EA) software can be used for viewing and changing the controller configuration parameters.

Axiomatic provides PC-based Electronic Assistant (EA) software to communicate with a wide range of Axiomatic products. The software can be downloaded from the Axiomatic website www.axiomatic.com.

The EA uses the Axiomatic USB-CAN converter P/N AX070501 to connect to the CAN network. The converter with cables can be ordered as an EA kit P/N AX070502 or AX070506K.

Please, refer to the user manual UMAX07050X for the description of the EA and associated products, and for the CAN network connection troubleshooting.

The user should use EA software version 5.15.113.0 or higher, which supports this controller firmware. The most recent EA software version can be downloaded from the Axiomatic website.

Before connecting to the CAN network, the user should ensure that the EA baud rate is the same as the baud rate used by ECUs on the network. The EA baud rate is displayed in the bottom-right corner of the EA screen and can be changed in the *Options* menu.

If the controller is the only one ECU on a temporary network set for configuring the unit, the EA baud rate should be set to the baud rate of the CAN network where the controller is planned to be deployed. This baud rate will be stored in the ECU non-volatile memory and used by the unit on the next power-up.

Upon connection, EA will show the controller on the list of ECUs that are present on the J1939 CAN network. If the controller is the only one ECU on the network, the following screen will appear, see Figure 11.

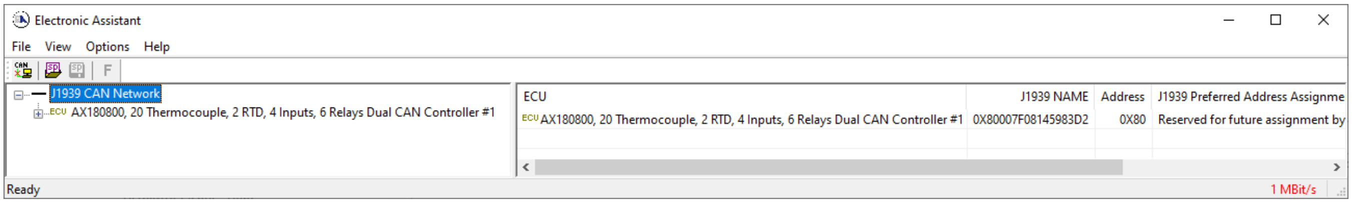


Figure 11. 20 Thermocouple, 2 RTD, 4 Inputs, 6 Relays Dual CAN Controller with SAE J1939 and Ethernet in EA

The user can then browse through the ECU parameters, read *General ECU Information* and *Bootloader Information* groups, view, and modify configuration parameters, see Figure 12.

The configuration parameters are grouped into function blocks. Please, refer to the appropriate section of this manual describing the required function block.

In the *General ECU Information* group, the user will see the version number of the application firmware. Please, make sure that the user manual version number matches with the most significant part of the application firmware version number. Otherwise, a different user manual is required to work with this controller.

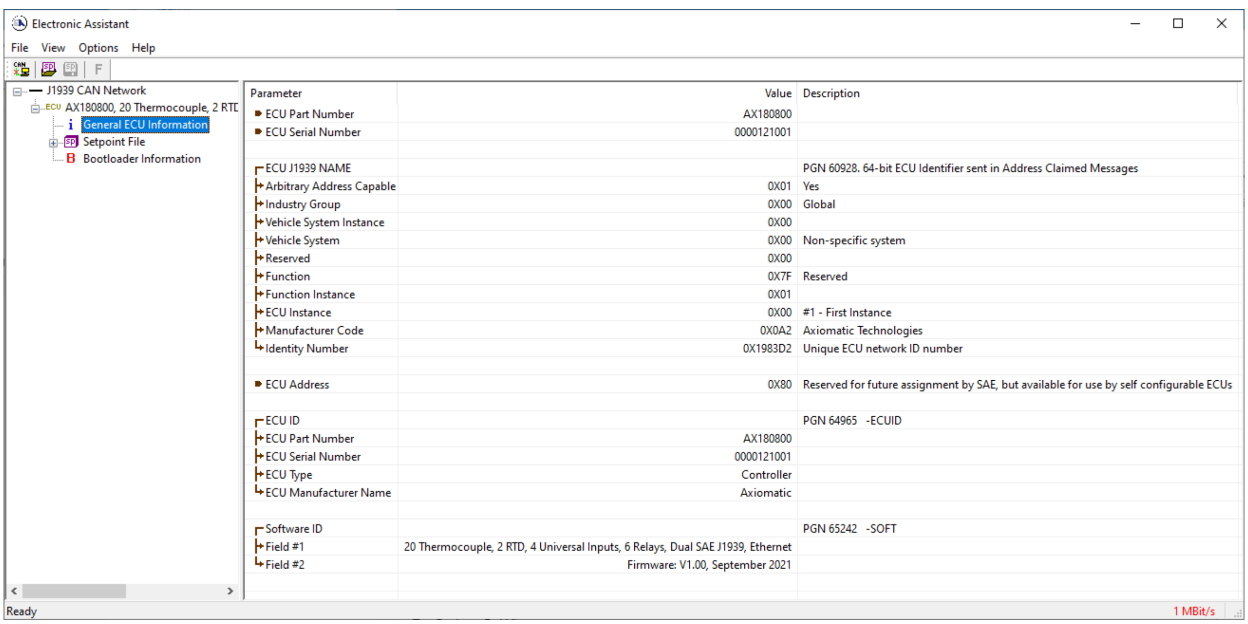


Figure 12. General ECU Information Screen

4.3 Function blocks in EA

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

4.3.1 J1939 Network Setpoints

The J1939 Network setpoints deal with the setpoints such as *ECU Instance Number* and *ECU Address*. Figure 13 and Table 37 below will explain these setpoints and their ranges.

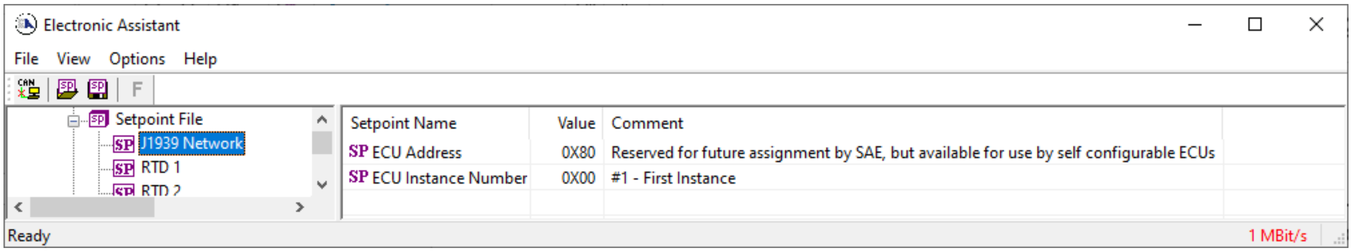


Figure 13: Screen Capture of Default J1939 Network Setpoints

Table 37: Default J1939 Network Setpoints

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

4.3.2 RTD Function Block

The RTD Input setpoints are defined in Section 3.5. Refer to that section for detailed information on how these setpoints are used. The screen capture below displays the available setpoints for each of the RTD Inputs.

Table 38 below highlights the allowable ranges for each setpoint. Due to the same default settings for each diagnostic, it will be mentioned in the table only once.

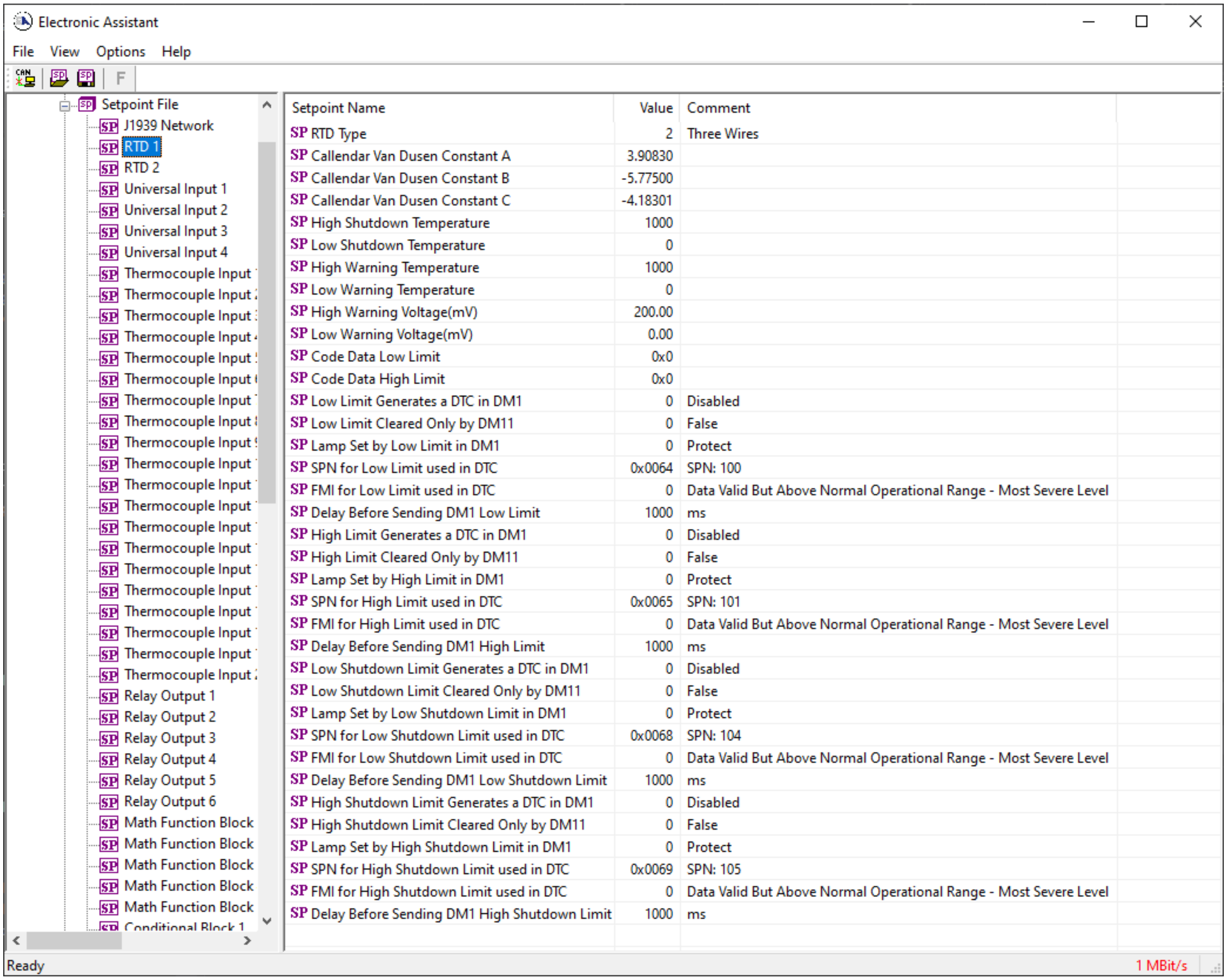


Figure 14: Screen Capture of Default RTD Input Setpoints

Table 38. Default RTD Setpoints

Name	Range	Default	Notes
RTD Type	Drop List	2, Three Wires	Refer to Section 3.5
Callendar Van Dusen Constant A	-100…100	3.90830	Refer to Section 3.5
Callendar Van Dusen Constant B	-100…100	-5.77500	Refer to Section 3.5
Callendar Van Dusen Constant C	-100…100	-4.18301	Refer to Section 3.5
High Shutdown Temperature	-2000…2000	1000	Refer to Section 3.5
Low Shutdown Temperature	-2000…2000	0	Refer to Section 3.5
High Warning Temperature	-2000…2000	1000	Refer to Section 3.5
Low Warning Temperature	-2000…2000	0	Refer to Section 3.5
High Warning Resistance	-20…200	200	Refer to Section 3.5
Low Warning Resistance	-20…200	0	Refer to Section 3.5
High Warning Code	0x00…0xFFFFFF	0	Refer to Section 3.5
Low Warning Code	0x00…0xFFFFFF	0	Refer to Section 3.5
Event Generates a DTC in DM1	Drop List	0, Disabled	Refer to Section 3.14
Event Cleared only by DM11	Drop List	0, False	Refer to Section 3.14
Lamp Set by Event in DTC	Drop List	0, Protect	Refer to Section 3.14
SPN for Event used in DTC	0x00…0x7FFFF	-	Refer to Section 3.14
FMI for Event used in DTC	Drop List	0	Refer to Section 3.14
Delay Before Sending DM1	0…60000	1000	Refer to Section 3.14

4.3.3 Universal Input Setpoints

The Universal Input setpoints are defined in Section 3.3.Refer to that section for detailed information on how these setpoints are used. The screen captures below in Figure 15 displays the available setpoints for each of the Universal Inputs.

Table 39 below highlights the allowable ranges for each setpoint. Due to the same default settings for each diagnostic, it will be mentioned in the table only once.

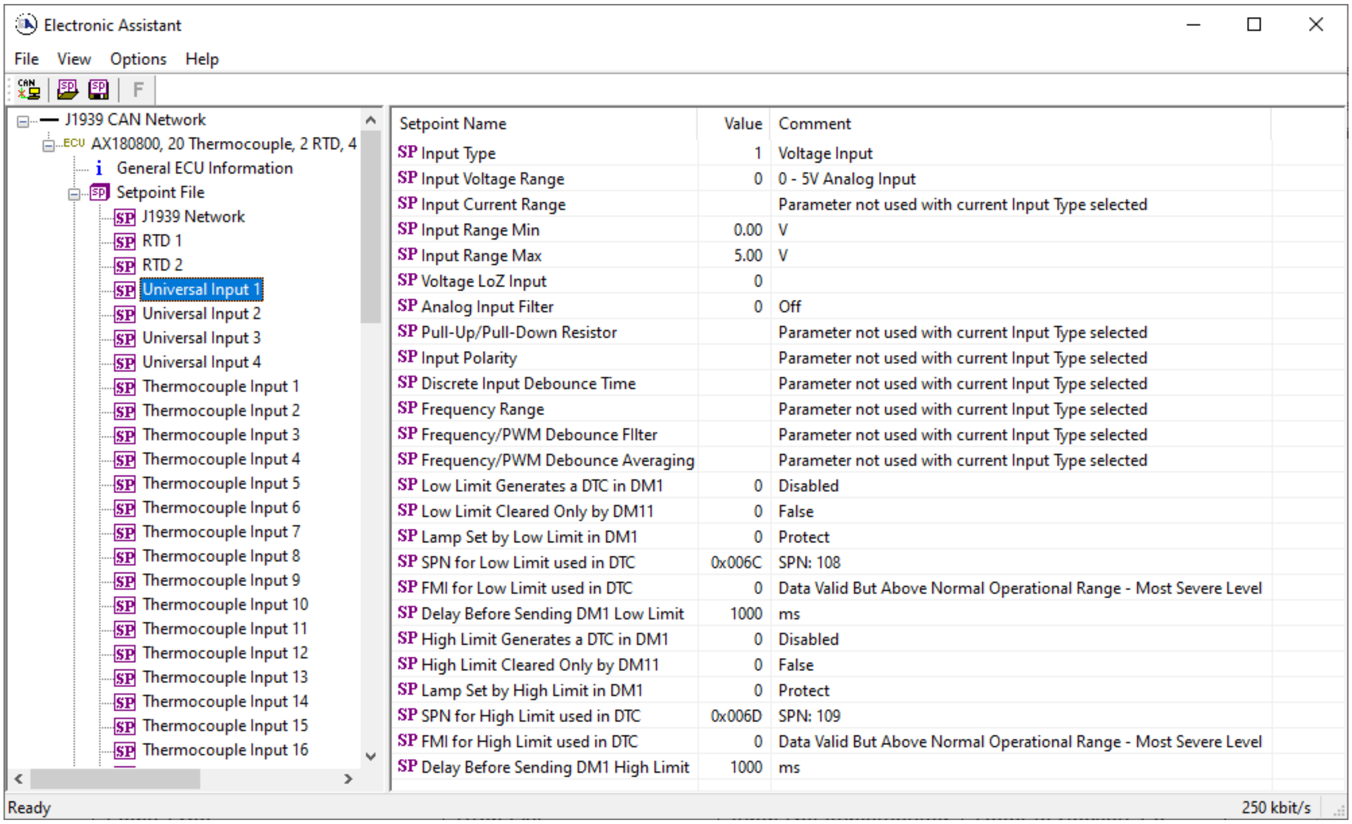


Figure 15: Screen Capture of Default Universal Input Setpoints

Table 39. Default Universal Input Setpoints

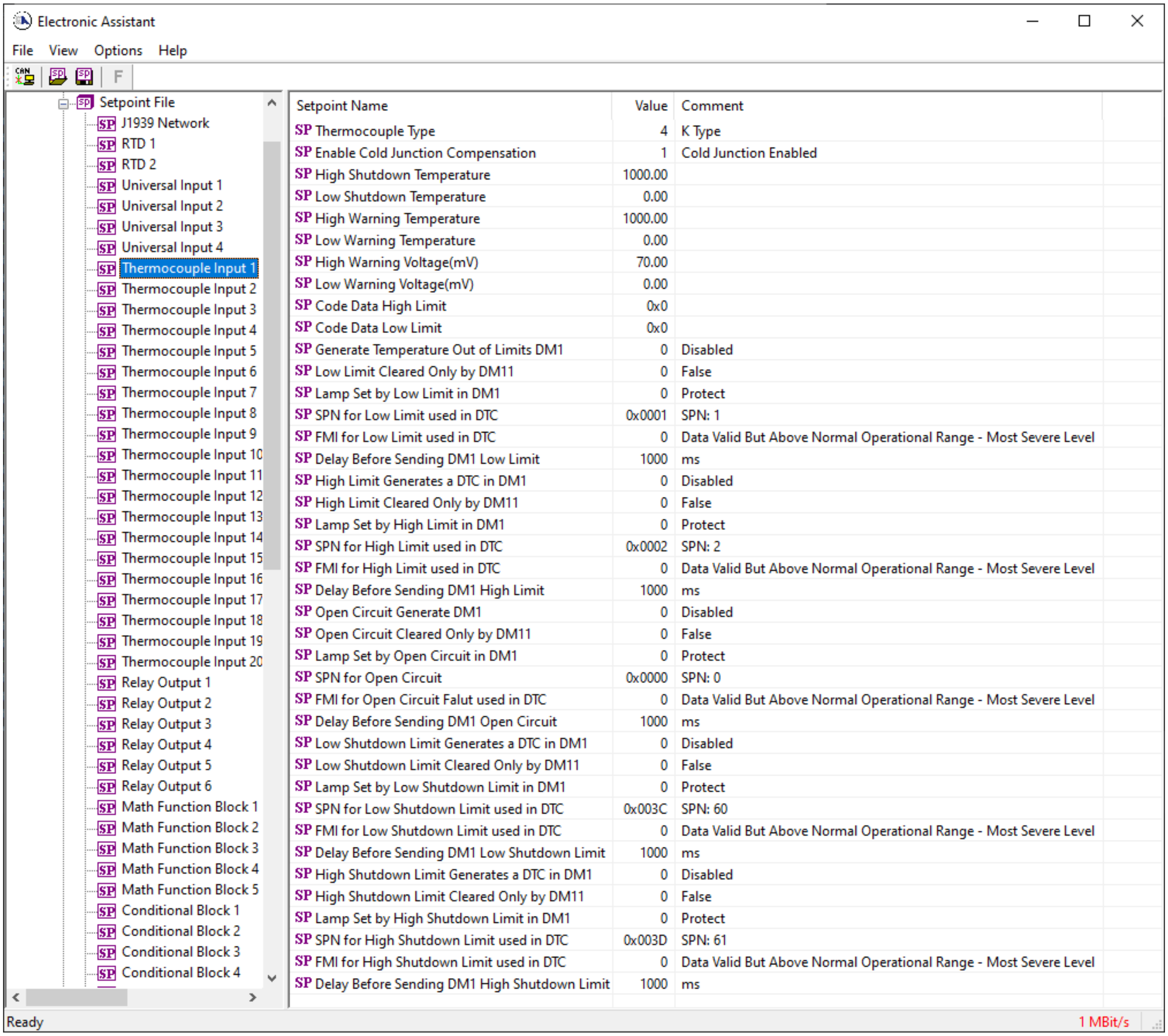
Name	**Range**	**Default**	**Notes**
Input Type	Drop List	1, Voltage Input	Refer to Section 3.3
Input Voltage Range	Drop List	0, 0-5V	Refer to Section 3.3
Input Current Range	Drop List	0, 0-20mA	Refer to Section 3.3
Input Range Min	Depends on Input Type	0	Refer to Section 3.3
Input Range Min	Depends on Input Type	5	Refer to Section 3.3
Voltage LoZ Input	Drop List	0, Pull Down Disabled	Refer to Section 3.3
Analog Input Filter	Drop List	Off	Refer to Section 3.3
Pull-Up/Pull-Down Resistor	Drop List	0, No Pull	Refer to Section 3.3
Input Polarity	Drop List	0, Active High	Refer to Section 3.3
Discrete Input Debounce Time	0…65000	50	Refer to Section 3.3
Frequency Range	0	0, 1Hz…10kHz	Refer to Section 3.3
Frequency/PWM Debounce Filter	Drop List	0, No Filter	Refer to Section 3.3
Frequency/PWM Debounce Averaging	Drop List	Input Not Implemented	Refer to Section 3.3
Event Generates a DTC in DM1	Drop List	0, Disabled	Refer to Section 3.14

Event Cleared only by DM11	Drop List	0, False	Refer to Section 3.14
Lamp Set by Event in DTC	Drop List	0, Protect	Refer to Section 3.14
SPN for Event used in DTC	0x00…0x7FFFF	-	Refer to Section 3.14
FMI for Event used in DTC	Drop List	0	Refer to Section 3.14
Delay Before Sending DM1	0…60000	1000	Refer to Section 3.14

4.3.4 Thermocouple Input Setpoints

The Thermocouple Input setpoints are defined in Section 3.4.Refer to that section for detailed information on how these setpoints are used. The screen captures below in Figure 16 displays the available setpoints for each of the Thermocouple Inputs.

Table 40 below highlights the allowable ranges for each setpoint.



Setpoint Name	Value	Comment
SP Thermocouple Type	4	K Type
SP Enable Cold Junction Compensation	1	Cold Junction Enabled
SP High Shutdown Temperature	1000.00	
SP Low Shutdown Temperature	0.00	
SP High Warning Temperature	1000.00	
SP Low Warning Temperature	0.00	
SP High Warning Voltage(mV)	70.00	
SP Low Warning Voltage(mV)	0.00	
SP Code Data High Limit	0x0	
SP Code Data Low Limit	0x0	
SP Generate Temperature Out of Limits DM1	0	Disabled
SP Low Limit Cleared Only by DM11	0	False
SP Lamp Set by Low Limit in DM1	0	Protect
SP SPN for Low Limit used in DTC	0x0001	SPN: 1
SP FMI for Low Limit used in DTC	0	Data Valid But Above Normal Operational Range - Most Severe Level
SP Delay Before Sending DM1 Low Limit	1000	ms
SP High Limit Generates a DTC in DM1	0	Disabled
SP High Limit Cleared Only by DM11	0	False
SP Lamp Set by High Limit in DM1	0	Protect
SP SPN for High Limit used in DTC	0x0002	SPN: 2
SP FMI for High Limit used in DTC	0	Data Valid But Above Normal Operational Range - Most Severe Level
SP Delay Before Sending DM1 High Limit	1000	ms
SP Open Circuit Generate DM1	0	Disabled
SP Open Circuit Cleared Only by DM11	0	False
SP Lamp Set by Open Circuit in DM1	0	Protect
SP SPN for Open Circuit	0x0000	SPN: 0
SP FMI for Open Circuit Falut used in DTC	0	Data Valid But Above Normal Operational Range - Most Severe Level
SP Delay Before Sending DM1 Open Circuit	1000	ms
SP Low Shutdown Limit Generates a DTC in DM1	0	Disabled
SP Low Shutdown Limit Cleared Only by DM11	0	False
SP Lamp Set by Low Shutdown Limit in DM1	0	Protect
SP SPN for Low Shutdown Limit used in DTC	0x003C	SPN: 60
SP FMI for Low Shutdown Limit used in DTC	0	Data Valid But Above Normal Operational Range - Most Severe Level
SP Delay Before Sending DM1 Low Shutdown Limit	1000	ms
SP High Shutdown Limit Generates a DTC in DM1	0	Disabled
SP High Shutdown Limit Cleared Only by DM11	0	False
SP Lamp Set by High Shutdown Limit in DM1	0	Protect
SP SPN for High Shutdown Limit used in DTC	0x003D	SPN: 61
SP FMI for High Shutdown Limit used in DTC	0	Data Valid But Above Normal Operational Range - Most Severe Level
SP Delay Before Sending DM1 High Shutdown Limit	1000	ms

Figure 16: Screen Capture of Default Thermocouple Input Setpoints

Table 40: Default Thermocouple Input Setpoints

Name	Range	Default	Notes
Thermocouple Type	Drop List	4, K Type	Refer to Section 3.4
Enable Cold Junction Compensation	Drop List	1	Refer to Section 3.4
High Shutdown Temperature	-2000…2000	1000	Refer to Section 3.4
Low Shutdown Temperature	-2000…2000	0	Refer to Section 3.4
High Warning Temperature	-2000…2000	1000	Refer to Section 3.4
Low Warning Temperature	-2000…2000	0	Refer to Section 3.4
High Warning Voltage	-78…78	70	Refer to Section 3.4
Low Warning Voltage	-78…78	0	Refer to Section 3.4
Code Data Low Limit	0x00...0xFFFFFF	0	Refer to Section 3.4
Code Data High Limit	0x00...0xFFFFFF	0	Refer to Section 3.4
Event Generates a DTC in DM1	Drop List	0, Disabled	Refer to Section 3.14
Event Cleared only by DM11	Drop List	0, False	Refer to Section 3.14
Lamp Set by Event in DTC	Drop List	0, Protect	Refer to Section 3.14
SPN for Event used in DTC	0x00…0x7FFFF	-	Refer to Section 3.14
FMI for Event used in DTC	Drop List	0	Refer to Section 3.14
Delay Before Sending DM1	0…60000	1000	Refer to Section 3.14

4.3.5 Relay Output Setpoints

The Relay Output setpoints are defined in Section 3.6.Refer to that section for detailed information on how these setpoints are used. The screen captures below in Figure 17 displays the available setpoints for each of the Thermocouple Inputs. Table 41 below highlights the allowable ranges for each setpoint.

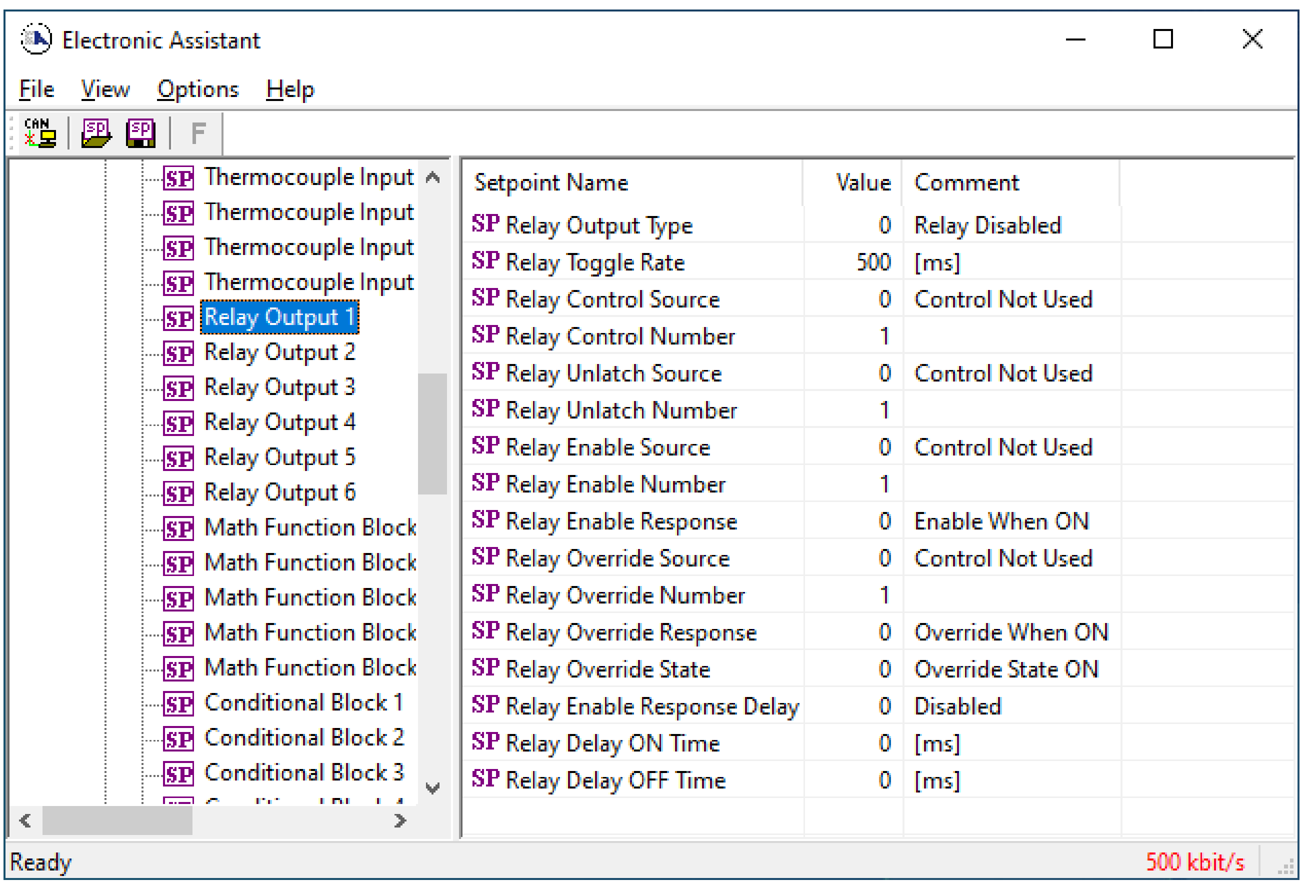


Figure 17: Screen Capture of Default Setpoints

Table 41. Default Relay Output Setpoints

Name	Range	Default	Notes
Relay Output Type	Drop List	0, Output Disabled	Refer to Section 3.6
Relay Toggle Rate	0…60000	500	Refer to Section 3.6
Relay Control Source	Drop List	0, Control Not Used	Refer to Table 5
Relay Control Number	Depends on Control Source	1	Refer to Table 5
Relay Unlatch Source	Drop List	0, Control Not Used	Refer to Table 5
Relay Unlatch Number	Depends on Control Source	1	Refer to Table 5
Relay Enable Source	Drop List	0, Control Not Used	Refer to Table 5
Relay Enable Number	Depends on Control Source	1	Refer to Table 5
Relay Enable Response	Drop List	0, Enable When ON	Refer to Section 3.6
Relay Override Source	Drop List	0, Control Not Used	Refer to Table 5
Relay Override Number	Depends on Control Source	1	Refer to Table 5
Relay Override Response	Drop List	0, Enable When ON	Refer to Section 3.6
Relay Override State	Drop List	0, Override State ON	Refer to Section 3.6
Relay Response Delay	Drop List	0, Disabled	Refer to Section 3.6
Relay Delay ON Time	0…60000	0	Refer to Section 3.6
Relay Delay OFF Time	0…60000	0	Refer to Section 3.6

4.3.6 Math Function Block Setpoints

The Relay Output setpoints are defined in Section 3.7.Refer to that section for detailed information on how these setpoints are used. The screen captures below in Figure 18 displays the available setpoints for each of the Thermocouple Inputs. Table *42* below highlights the allowable ranges for each setpoint.

Please note: To show the setpoints, the **Math Enabled setpoint was changed from its default value.*

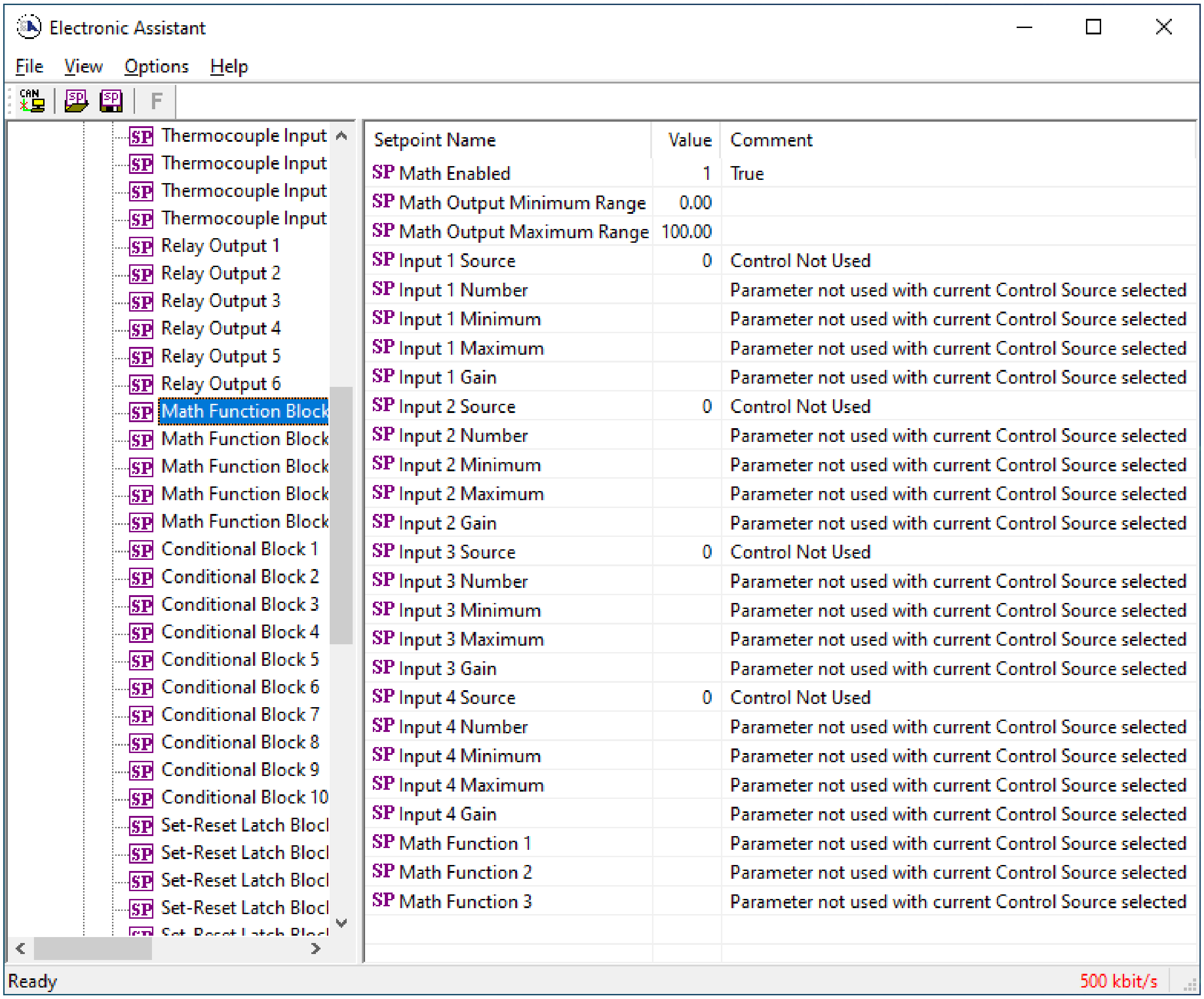


Figure 18: Screen Capture of Math Default Setpoints

Table 42: Default Math Function Block Setpoints

Name	Range	Default	Notes
Math Enabled	Drop List	False	
Math Output Minimum Range	-32768…32767	0	
Math Output Maximum Range	-32768…32767	100.0	
Input 1 Source	Drop List	Control not used	Refer to Table 5
Input 1 Number	Depends on control source	1	Refer to Table 5
Input 1 Minimum	$-10^6 \ldots 10^6$	0.00	
Input 1 Maximum	$-10^6 \ldots 10^6$	100.00	
Input 1 Gain	-100…100	1	
Input 2 Source	Drop List	Control not used	Refer to Table 5
Input 2 Number	Depends on control source	1	Refer to Table 5
Input 2 Minimum	$-10^6 \ldots 10^6$	0.00	
Input 2 Maximum	$-10^6 \ldots 10^6$	100.00	
Input 2 Gain	-100…100	1	
Input 3 Source	Drop List	Control not used	Refer to Table 5
Input 3 Number	Depends on control source	1	Refer to Table 5
Input 3 Minimum	$-10^6 \ldots 10^6$	0.00	
Input 3 Maximum	$-10^6 \ldots 10^6$	100.00	

Input 3 Gain	-100…100	1	
Input 4 Source	Drop List	Control not used	Refer to Table 5
Input 4 Number	Depends on control source	1	Refer to Table 5
Input 4 Minimum	$-10^6 \ldots 10^6$	0.00	
Input 4 Maximum	$-10^6 \ldots 10^6$	100.00	
Input 4 Gain	-100…100	1	
Math Function 1	Drop List	=, True When InA Equals InB	SeeTable *22. Math function X Operator Options*
Math Function 2	Drop List	=, True When InA Equals InB	SeeTable *22. Math function X Operator Options*
Math Function 3	Drop List	=, True When InA Equals InB	SeeTable *22. Math function X Operator Options*

4.3.7 Conditional Logic Block Setpoints

The Conditional Block setpoints are defined in Section 3.8. Refer to that section for detailed information on how these setpoints are used. The screen capture in Figure 19 displays the available setpoints for each of the Conditional Blocks. The table below the screen capture highlights the allowable ranges for each setpoint.

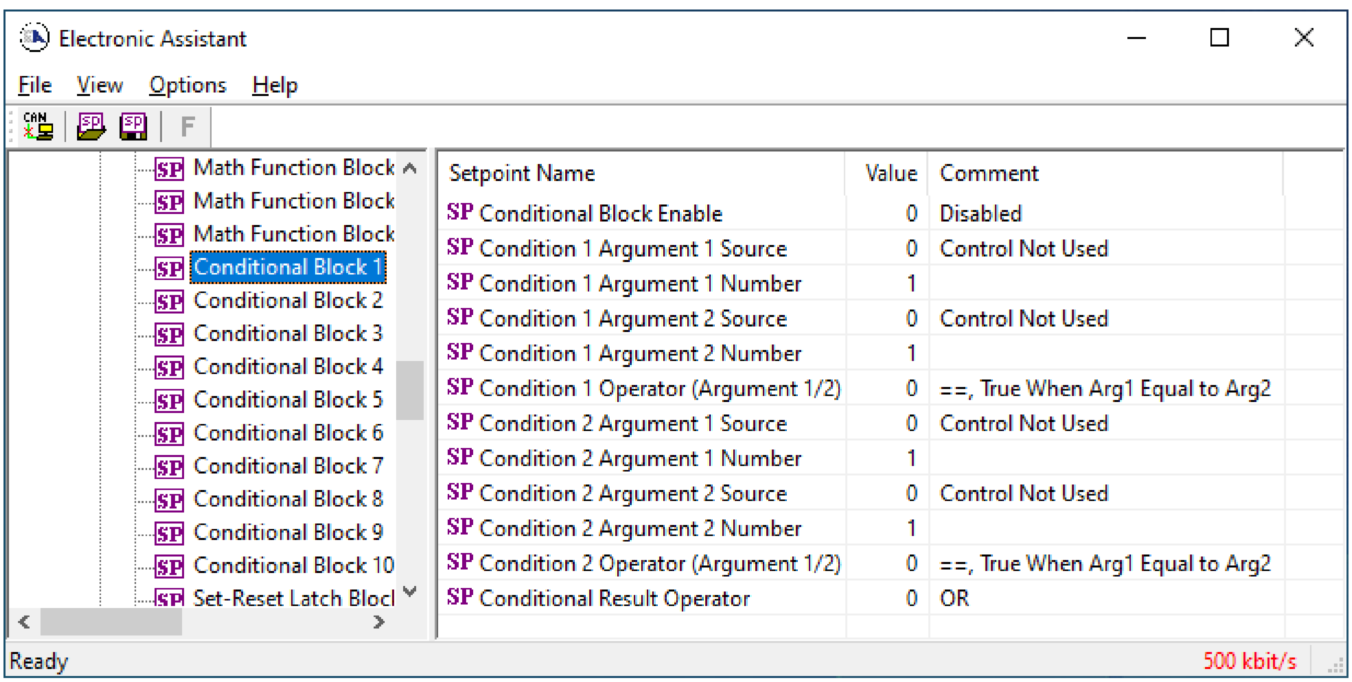


Figure 19: Screen Capture of Conditional Block Setpoints

Table 43. Default Conditional Block Setpoints

Name	Range	Default	Notes
Conditional Function Enabled	Drop List	Disabled	
Condition 1 Argument 1 Source	Drop List	Digital Input	Refer to Table 5
Condition 1 Argument 1 Number	Depends on Source Selected	0	Refer to Table 5

Condition 1 Argument 2 Source	Drop List	Digital Input	Refer to Table 5
Condition 1 Argument 2 Number	Depends on Source Selected	0	Refer to Table 5
Condition 1 Operator (Argument 1/2)	Drop List	0	Refer to Table 23
Condition 2 Argument 1 Source	Drop List	Digital Input	Refer to Table 5
Condition 2 Argument 1 Number	Depends on Source Selected	0	Refer to Table 5
Condition 2 Argument 2 Source	Drop List	Digital Input	Refer to Table 5
Condition 2 Argument 2 Number	Depends on Source Selected	0	Refer to Table 5
Condition 2 Operator (Argument 1/2)	Drop List	0	Refer to Table 23
Conditional Result Operator	Drop List	OR	Refer to Table 24

4.3.8 Set-Reset Latch Block

The Set-Reset Latch Block setpoints are defined in Section 3.9. Refer to that section for detailed information on how these setpoints are used. The screen capture in Figure 20 displays the available setpoints for each of the Set-Reset Latch Blocks. The table below the screen capture highlights the allowable ranges for each setpoint.

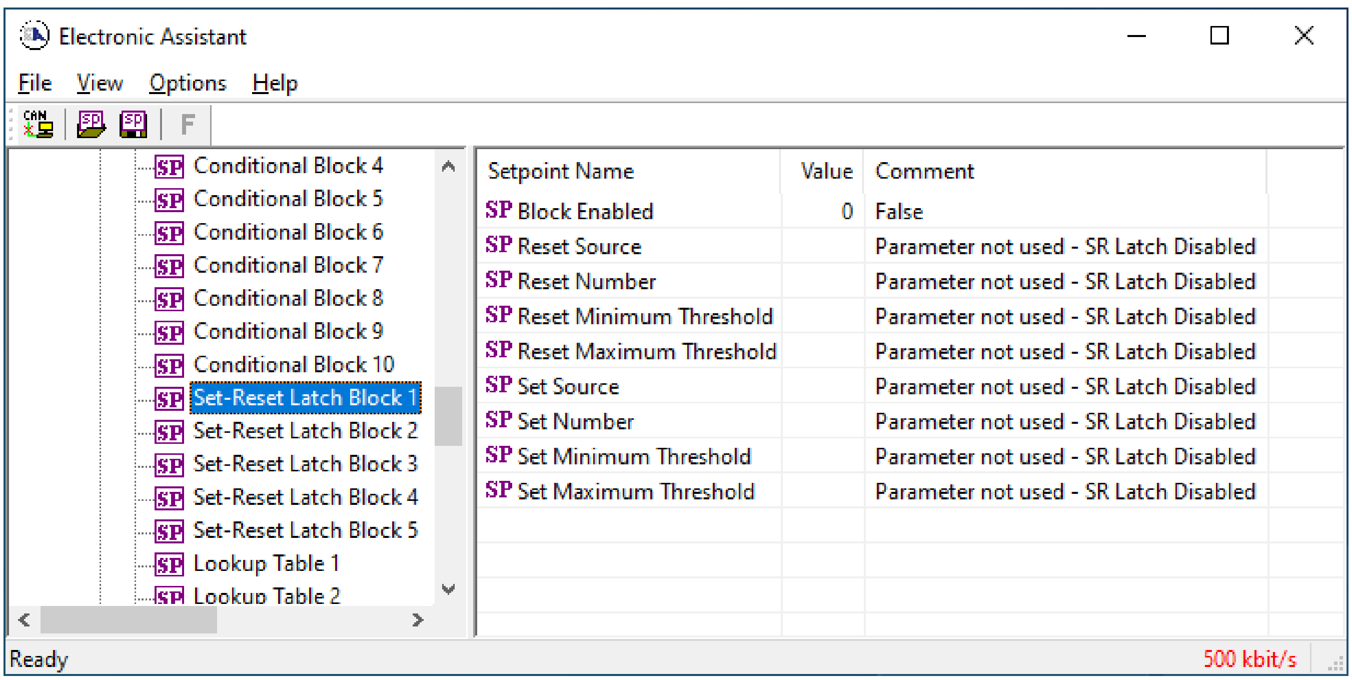


Figure 20: Screen Capture of Set-Reset Latch Block Setpoints

Table 44. Default Set-Reset Latch Block Setpoints

Name	Range	Default	Notes
Block Enabled	Drop List	False	
Reset Source	Drop List	Control Not Used	Refer to Table 5
Reset Number	Depends on Source Selected	1	Refer to Table 5
Reset Minimum Threshold	Drop List	0%	Refer to Section 3.9
Reset Maximum Threshold	Depends on Source Selected	100%	Refer to Section 3.9
Set Source	Drop List	Control Not Used	Refer to Table 5
Set Number	Drop List	1	Refer to Table 5
Set Minimum Threshold	Depends on Source Selected	0%	Refer to Section 3.9
Set Maximum Threshold	Drop List	100%	Refer to Section 3.9

4.3.9 Lookup Table Setpoints

The Lookup Table Block setpoints are defined in Section 3.10. Refer to that section for detailed information on how these setpoints are used. The screen capture in Figure 21 displays the available setpoints for each of the Lookup Table Setpoints. The table below the screen capture highlights the allowable ranges for each setpoint.
Please note: To show the setpoints, the **X-Axis Source setpoint was changed from its default value.*

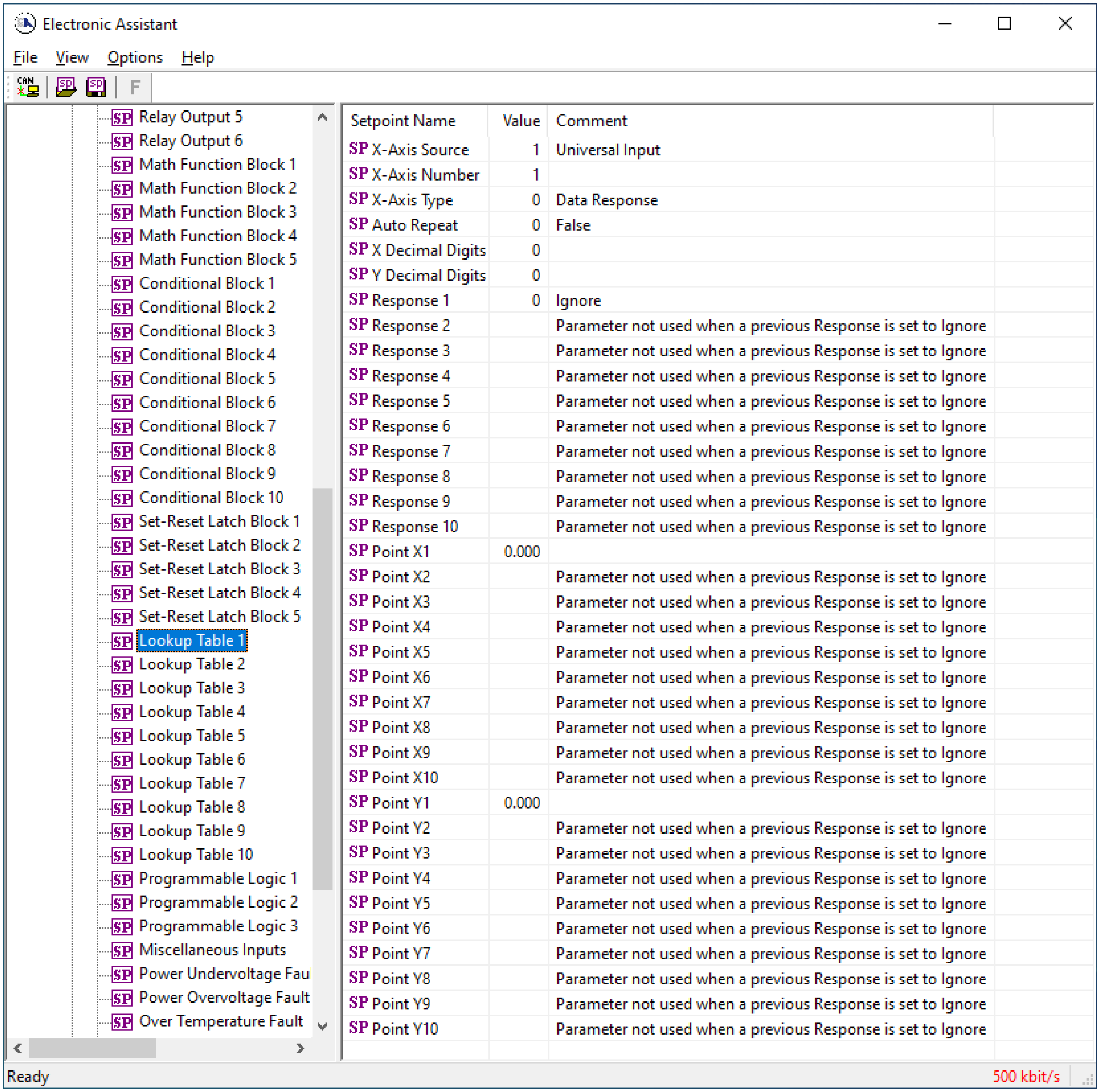


Figure 21: Screen Capture of Lookup Table Setpoints

Table 45. Default Lookup Table Setpoints

Name	**Range**	**Default**	**Notes**
X-Axis Source	Drop List	Control Not Used	Refer to Table 5
X-Axis Number	Depends on control source	1	Refer to Table 5
X-Axis Type	Drop List	Data Response	See Table 26
Auto Repeat	Drop List	False	
X Decimal Digits	0…3	0	Resolution is 10^x, affects X points
Y Decimal Digits	0…3	0	Resolution is 10^x, affects Y points
Response 1	Drop List	Ramp To	See Table 27
Response 2	Drop List	Ramp To	See Table 27
Response 3	Drop List	Ramp To	See Table 27
Response 4	Drop List	Ramp To	See Table 27
Response 5	Drop List	Ramp To	See Table 27
Response 6	Drop List	Ramp To	See Table 27
Response 7	Drop List	Ramp To	See Table 27
Response 8	Drop List	Ramp To	See Table 27
Response 9	Drop List	Ramp To	See Table 27
Response 10	Drop List	Ramp To	See Table 27
Point X1	From X-Axis source minimum to Point 1 - X Value	X-Axis source minimum Depends on the Table number	See Section 3.10
Point X2	From Point 0 - X Value to Point 2 - X Value	Depends on the Table number	See Section 3.10
Point X3	From Point 1 - X Value to Point 3 - X Value	Depends on the Table number	See Section 3.10
Point X4	From Point 2 - X Value to Point 4 - X Value	Depends on the Table number	See Section 3.10
Point X5	From Point 3 - X Value to Point 5 - X Value source	Depends on the Table number	See Section 3.10
Point X6	From Point 4 - X Value to Point 6 - X Value	Depends on the Table number	See Section 3.10
Point X7	From Point 5 - X Value to Point 7 - X Value	Depends on the Table number	See Section 3.10
Point X8	From Point 6 - X Value to Point 8 - X Value	Depends on the Table number	See Section 3.10
Point X9	From Point 7 - X Value to Point 9 - X Value	Depends on the Table number	See Section 3.10
Point X10	From Point 8 - X Value to Point 10 - X Value	Depends on the Table number	See Section 3.10
Point Y1	-10^6 to 10^6	Depends on the Table number	
Point Y2	-10^6 to 10^6	Depends on the Table number	
Point Y3	-10^6 to 10^6	Depends on the Table number	
Point Y4	-10^6 to 10^6	Depends on the Table number	
Point Y5	-10^6 to 10^6	Depends on the Table number	
Point Y6	-10^6 to 10^6	Depends on the Table number	
Point Y7	-10^6 to 10^6	Depends on the Table number	
Point Y8	-10^6 to 10^6	Depends on the Table number	
Point Y9	-10^6 to 10^6	Depends on the Table number	
Point Y10	-10^6 to 10^6	Depends on the Table number	

4.3.10 Programmable Logic Block Setpoints

The Programmable Logic Block setpoints are defined in Section 3.11. Refer to that section for detailed information on how these setpoints are used. The screen capture in Figure 22 displays the available setpoints for each of the Programmable Logic Blocks. The table below the screen capture highlights the allowable ranges for each setpoint.
Please note: To show the setpoints, the **Logic Enabled setpoint was changed from its default value.*

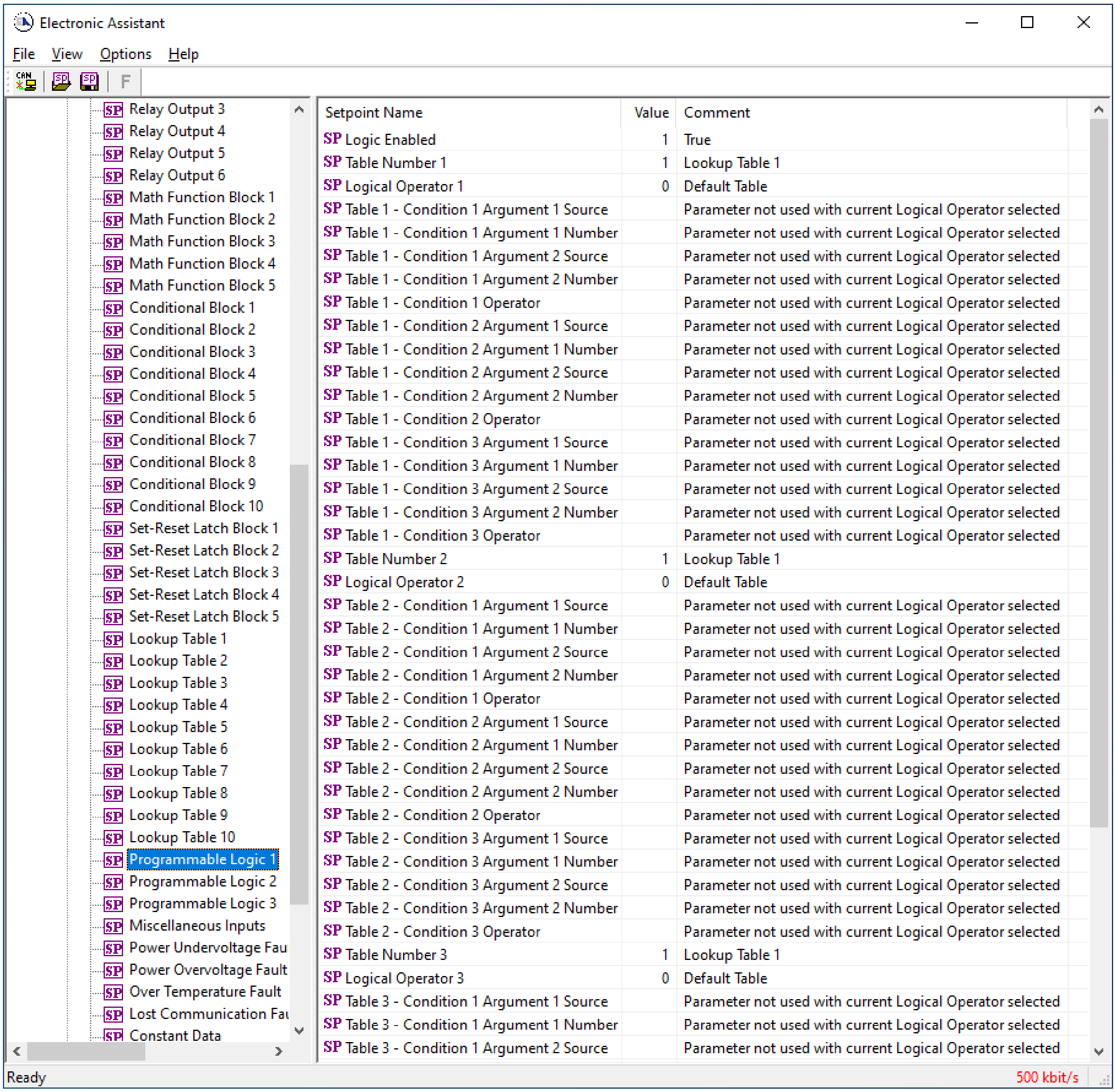


Figure 22: Screen Capture of Programmable Logic Block Setpoints

Table 46. Default Programmable Logic Block Setpoints

Name	Range	Default	Notes
Logic Enabled	Drop List	False	
Table Number 1	1 to 10	Lookup Table 1	
Logical Operator 1	Drop List	Default Table	See Table 29
Table 1 - Condition 1 Argument 1 Source	Drop List	Control Not Used	Refer to Table 5
Table 1 - Condition 1 Argument 1 Number	Depends on control source	1	Refer to Table 5
Table 1 - Condition 1 Argument 2 Source	Drop List	Control Not Used	Refer to Table 5
Table 1 - Condition 1 Argument 2 Number	Depends on control source	1	Refer to Table 5
Table 1 - Condition 1 Operator	Drop List	=, Equal	See Table 28
Table 1 - Condition 2 Argument 1 Source	Drop List	Control Not Used	Refer to Table 5
Table 1 - Condition 2 Argument 1 Number	Depends on control source	1	Refer to Table 5
Table 1 - Condition 2 Argument 2 Source	Drop List	Control Not Used	Refer to Table 5
Table 1 - Condition 2 Argument 2 Number	Depends on control source	1	
Table 1 - Condition 2 Operator	Drop List	=, Equal	See Table 28
Table 1 - Condition 3 Argument 1 Source	Drop List	Control Not Used	Refer to Table 5
Table 1 - Condition 3 Argument 1 Number	Depends on control source	1	Refer to Table 5
Table 1 - Condition 3 Argument 2 Source	Drop List	Control Not Used	Refer to Table 5
Table 1 - Condition 3 Argument 2 Number	Depends on control source	1	Refer to Table 5
Table 1 - Condition 3 Operator	Drop List	=, Equal	See Table 28

4.3.11 Miscellaneous Setpoints

The Miscellaneous setpoints are defined in Section 3.13. Refer to that section for detailed information on how these setpoints are used. The screen captures below in Figure 23 displays the available setpoints.
Table *47* highlights the allowable ranges for each setpoint.

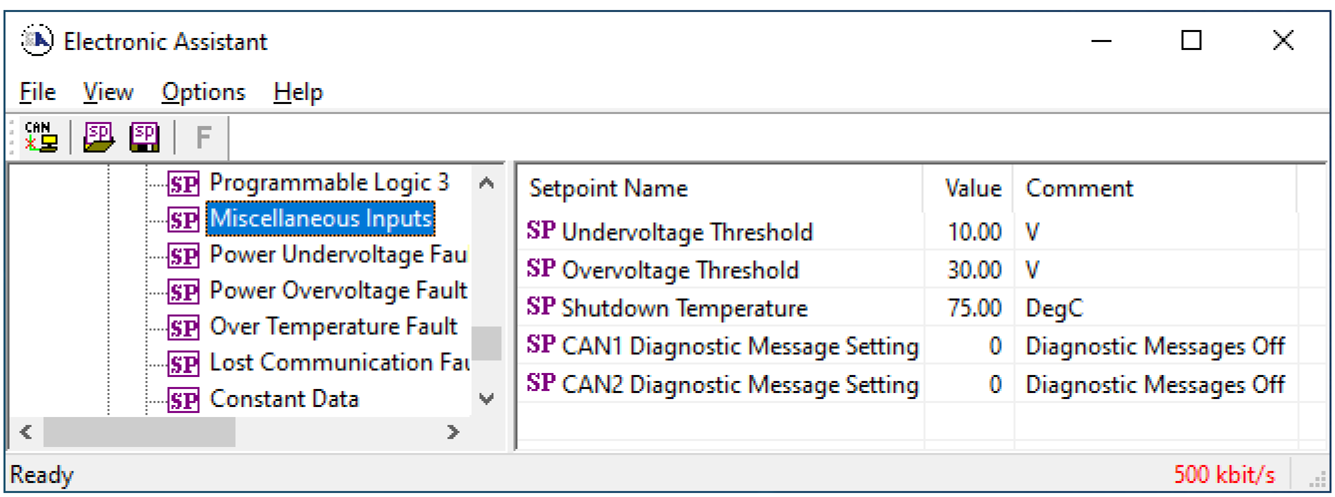


Figure 23: Screen Capture of Default Miscellaneous Setpoints

Table 47. Default Miscellaneous Setpoints

Name	Range	Default	Notes
Undervoltage Threshold	6.0…36.0	10.0	Units in [Volts]
Overvoltage Threshold	6.0…36.0	30.0	Units in [Volts]
Shutdown Temperature	40…125	75	Units in [Celsius]
CAN1 Diagnostic Message Setting	Drop List	Diagnostics Messages Off	
CAN2 Diagnostic Message Setting	Drop List	Diagnostics Messages Off	

4.3.12 Diagnostic Setpoints

The Diagnostic setpoints are defined in Section 3.14. Refer to that subsection for detailed information on how these setpoints are used. The screen captures below in Figure 24 displays the available setpoints for the Power Supply Diagnostic setpoints. Table 48 below highlights the allowable ranges for each setpoint.

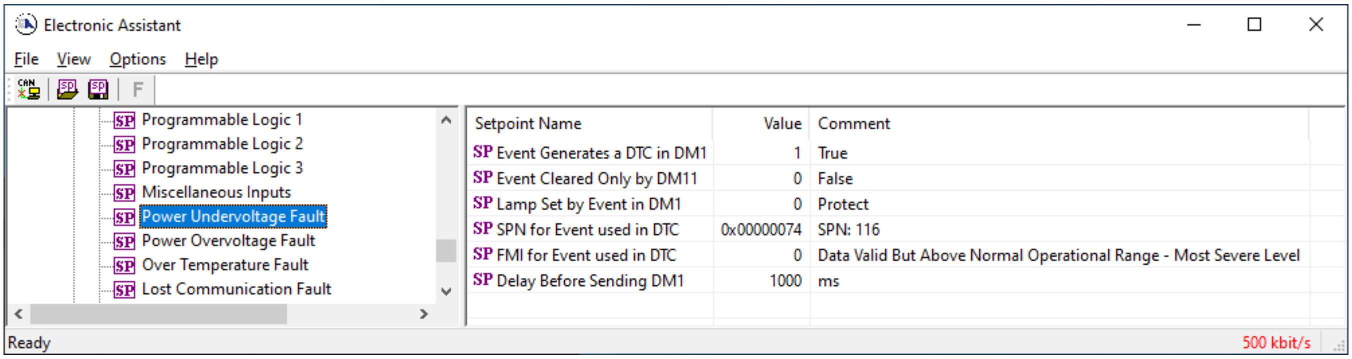


Figure 24: Screen Capture of Default Diagnostic Setpoints

Table 48. Default Values of Diagnostic Setpoints

Name	**Range**	**Default**	**Notes**
Event Generates a DTC in DM1	Drop List	False	Default changed to *True* for illustration purposes, Refer to Section 3.14
Event Cleared Only by DM11	Drop List	False	Refer to Section 3.14
Lamp Set by Event in DM1	Drop List	Amber, Warning	Refer to Section 3.14
SPN for Event used in DTC	0...524,287	-	Refer to Section 3.14
FMI for Event used in DTC	Drop List	Voltage Below Normal, Or Shorted to Low Source	Refer to Table 31
Delay Before Sending DM1	0…60000	1000	Units in [milliseconds]

4.3.13 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. The two constants are fully user configurable to any value between +/- 1,000,000. The default values are displayed in the screen capture below.

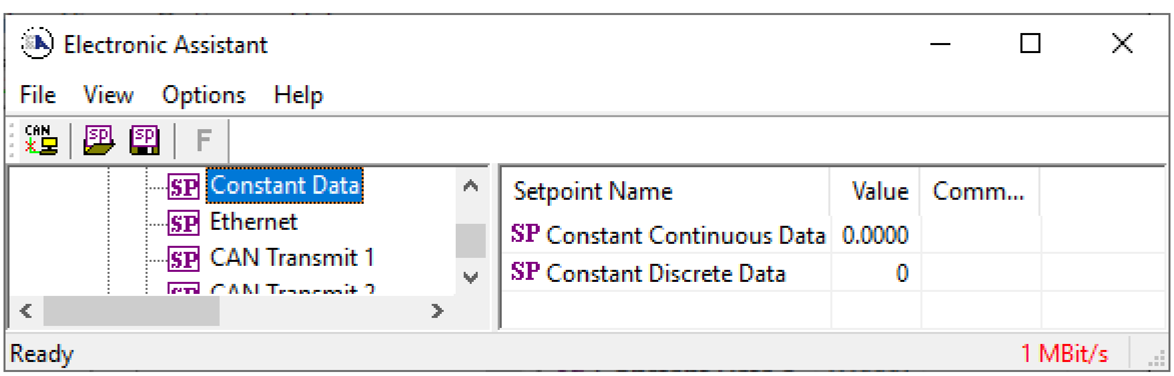


Figure 25: Screen Capture of Constant Data Setpoints

4.3.14 Ethernet Setpoints

The Ethernet is explained in Section 3.17. The picture below shows available setpoints for the Ethernet function block. The Table 49 shows the default values of the Ethernet Function Block setpoints.

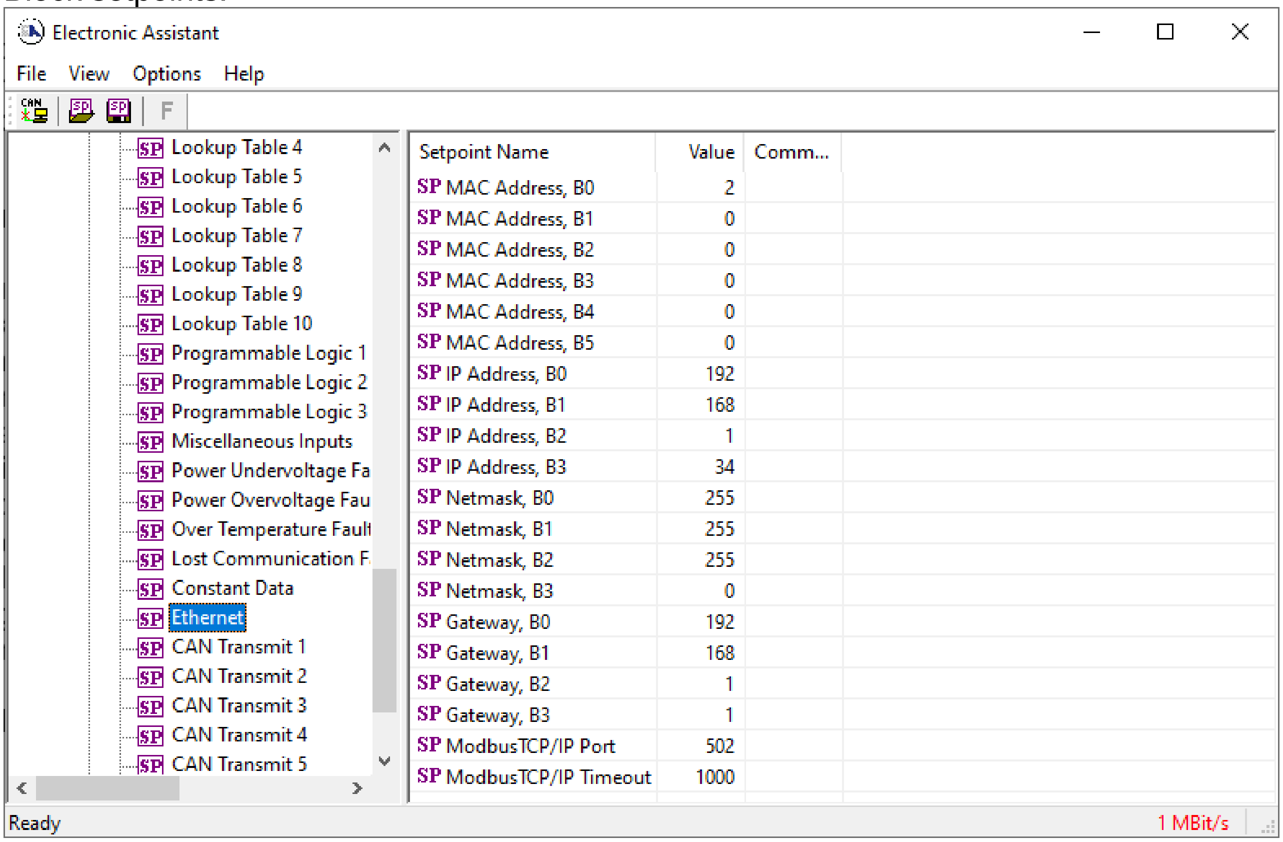


Figure 26: Screen Capture of Ethernet Setpoints

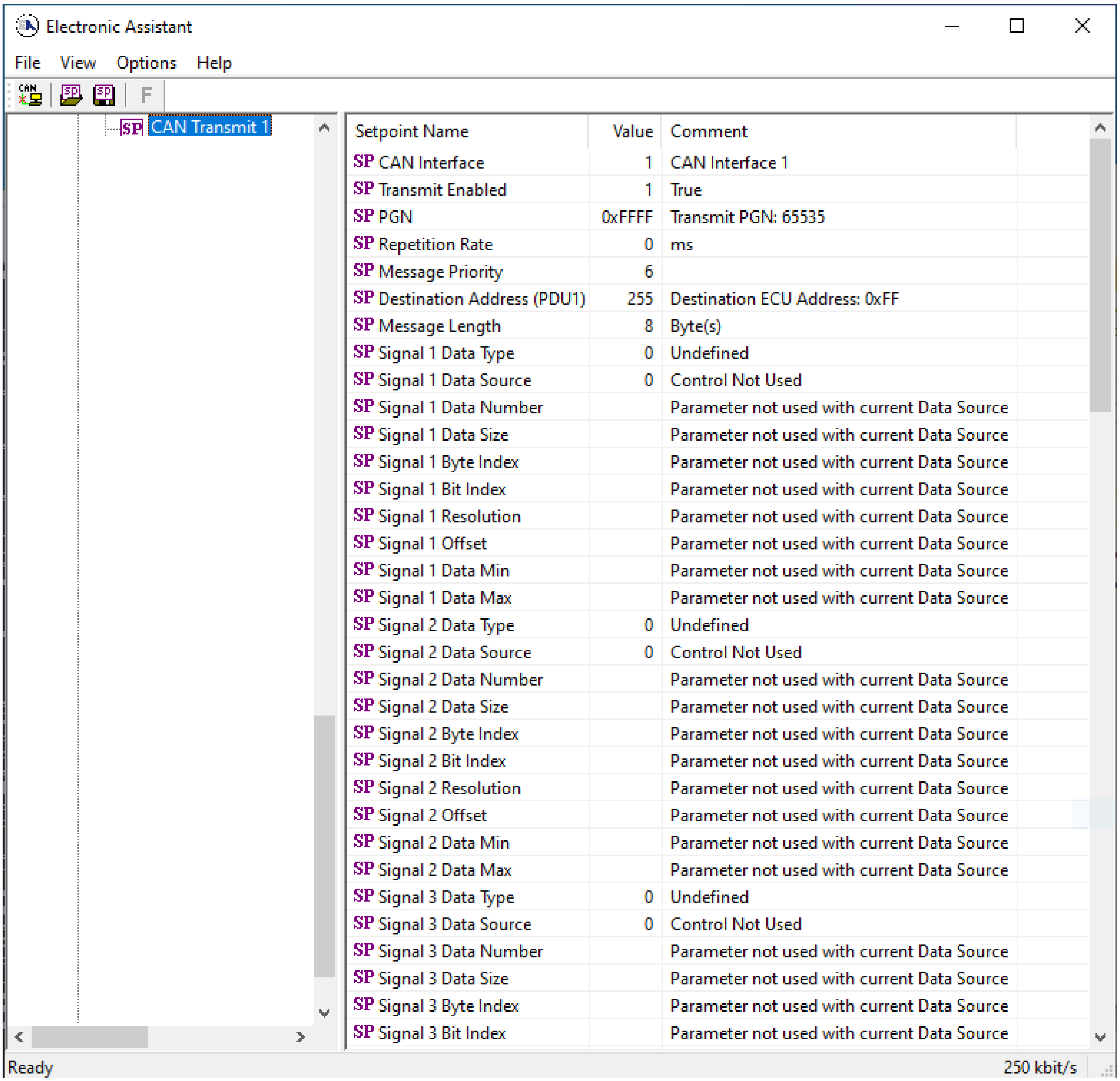
Table 49. Default Values of Ethernet Setpoints

Name	**Range**	**Default**	**Notes**
MAC Address, B0	0…0xFF	2	Refer to Section 3.17
MAC Address, B1	0…0xFF	0	Refer to Section 3.17
MAC Address, B2	0…0xFF	0	Refer to Section 3.17
MAC Address, B3	0…0xFF	0	Refer to Section 3.17
MAC Address, B4	0…0xFF	0	Refer to Section 3.17
MAC Address, B5	0…0xFF	0	Refer to Section 3.17
IP Address, B0	0…0xFF	192	Refer to Section 3.17
IP Address, B1	0…0xFF	168	Refer to Section 3.17
IP Address, B2	0…0xFF	1	Refer to Section 3.17
IP Address, B3	0…0xFF	34	Refer to Section 3.17
Netmask, B0	0…0xFF	255	Refer to Section 3.17
Netmask, B1	0…0xFF	255	Refer to Section 3.17
Netmask, B2	0…0xFF	255	Refer to Section 3.17
Netmask, B3	0…0xFF	0	Refer to Section 3.17
Gateway, B0	0…0xFF	192	Refer to Section 3.17
Gateway, B1	0…0xFF	168	Refer to Section 3.17
Gateway, B2	0…0xFF	1	Refer to Section 3.17

Gateway, B3	0…0xFF	1	Refer to Section 3.17
Modbus TCP/IP Port	65535	502	Refer to Section 3.17
Modbus TCP/IP Timeout	65535	1000	Refer to Section 3.17

4.3.15 CAN Transmit Setpoints

The performance of CAN Transmit is explained in Section 3.20. The picture below shows available CAN Transmit setpoints. The
**Please note: Even though there are only 4 signals presented on the screen capture, there are 10 signals for each CAN Transmit.*
*To show the setpoints, the **Transmit Enabled** setpoint was changed from its default value.*



Setpoint Name	Value	Comment
SP CAN Interface	1	CAN Interface 1
SP Transmit Enabled	1	True
SP PGN	0xFFFF	Transmit PGN: 65535
SP Repetition Rate	0	ms
SP Message Priority	6	
SP Destination Address (PDU1)	255	Destination ECU Address: 0xFF
SP Message Length	8	Byte(s)
SP Signal 1 Data Type	0	Undefined
SP Signal 1 Data Source	0	Control Not Used
SP Signal 1 Data Number		Parameter not used with current Data Source
SP Signal 1 Data Size		Parameter not used with current Data Source
SP Signal 1 Byte Index		Parameter not used with current Data Source
SP Signal 1 Bit Index		Parameter not used with current Data Source
SP Signal 1 Resolution		Parameter not used with current Data Source
SP Signal 1 Offset		Parameter not used with current Data Source
SP Signal 1 Data Min		Parameter not used with current Data Source
SP Signal 1 Data Max		Parameter not used with current Data Source
SP Signal 2 Data Type	0	Undefined
SP Signal 2 Data Source	0	Control Not Used
SP Signal 2 Data Number		Parameter not used with current Data Source
SP Signal 2 Data Size		Parameter not used with current Data Source
SP Signal 2 Byte Index		Parameter not used with current Data Source
SP Signal 2 Bit Index		Parameter not used with current Data Source
SP Signal 2 Resolution		Parameter not used with current Data Source
SP Signal 2 Offset		Parameter not used with current Data Source
SP Signal 2 Data Min		Parameter not used with current Data Source
SP Signal 2 Data Max		Parameter not used with current Data Source
SP Signal 3 Data Type	0	Undefined
SP Signal 3 Data Source	0	Control Not Used
SP Signal 3 Data Number		Parameter not used with current Data Source
SP Signal 3 Data Size		Parameter not used with current Data Source
SP Signal 3 Byte Index		Parameter not used with current Data Source
SP Signal 3 Bit Index		Parameter not used with current Data Source

Figure 27 Available Setpoint for CAN Transmit Function Block

Table 50. Default CAN Transmit Setpoints

Name	Range	Default	Notes
CAN interface	Drop List	1, CAN Interface 1	
PGN	0…65,535	65,280	Refer to Section 3.20
Repetition Rate	0…60,000	1000	Refer to Section 3.20
Message Priority	0...7	6	Refer to Section 3.20
Destination Address (PDU1)	0...255	254	Refer to Section 3.20
Data Source	Drop List	CAN Status Report	Refer to Table 5
Data Number	Depends on Source Selected	1	Refer to Table 5
Data Size	0…32	0 bits	Refer to Section 3.20
Pos Byte	Depends on Source Selected	0	Refer to Section 3.20
Pos Bit	Depends on Source Selected	65,280	Refer to Section 3.20
Resolution	-100,000...100,000	1	Refer to Section 3.20
Offset	-100,000...100,000	0	Refer to Section 3.20

4.3.16 CAN Receive Setpoints

The CAN Receive setpoints are defined in Section 3.19. Refer to that section for detailed information on how these setpoints are used. The screen capture below in Figure 28 displays the available setpoints for the CAN Receive setpoints. Table 51 below highlights the allowable ranges for each setpoint.

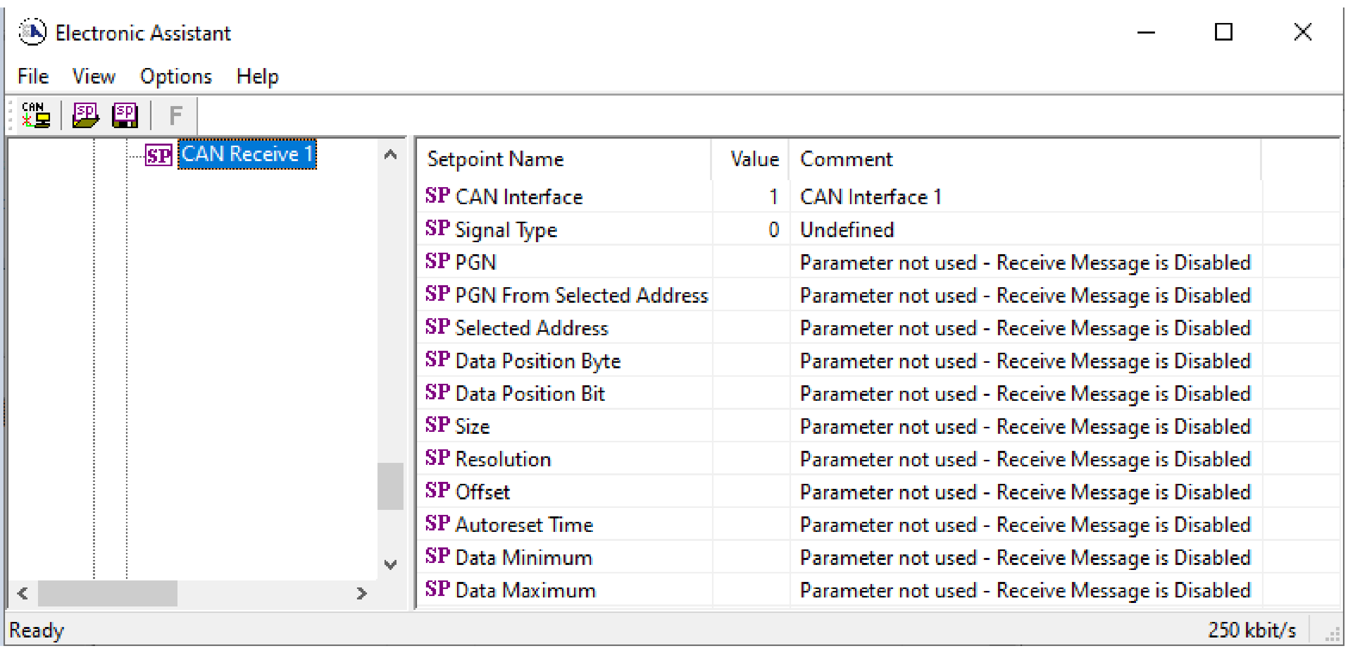


Figure 28: Screen Capture of Default CAN Receive Setpoints

Table 51. Default CAN Receive Setpoints

Name	Range	Default	Notes
CAN interface	Drop List	1, CAN Interface 1	Refer to Section 3.19
Signal Type	Drop List	0, Undefined	Default changed to *Contunious* for illustration purposes. Refer to Section 3.19
PGN	0…65,535	65,535	Refer to Section 3.19

PGN From Selected Address	Drop List	0, False	Refer to Section 3.19
Selected Address	0…255	1	Refer to Section 3.19
Data Position Byte	0…7	1	Refer to Section 3.19
Data Position Bit	0…7	1	Refer to Section 3.19
Size	0…32	0 bits	Refer to Section 3.19
Resolution	-0xFFFFFFFF…0xFFFFFFFF	1.0	Refer to Section 3.19
Offset	-0xFFFFFFFF…0xFFFFFFFF	0.0	Refer to Section 3.19
Autoreset Time	0...60,000	500	Refer to Section 3.19
Data Minimum	-0xFFFFFFFF…Data Max	0.0	Refer to Section 3.19
Data Maximum	Data Min…0xFFFFFFFF	100.0	Refer to Section 3.19

4.3.17 Modbus Receive Function Block

The Modbus Receive Function Block is defined in section 3.18. Please refer there for detailed information about how these setpoints are used.
Please note: some setpoints were changed from their original values for demonstration purposes.

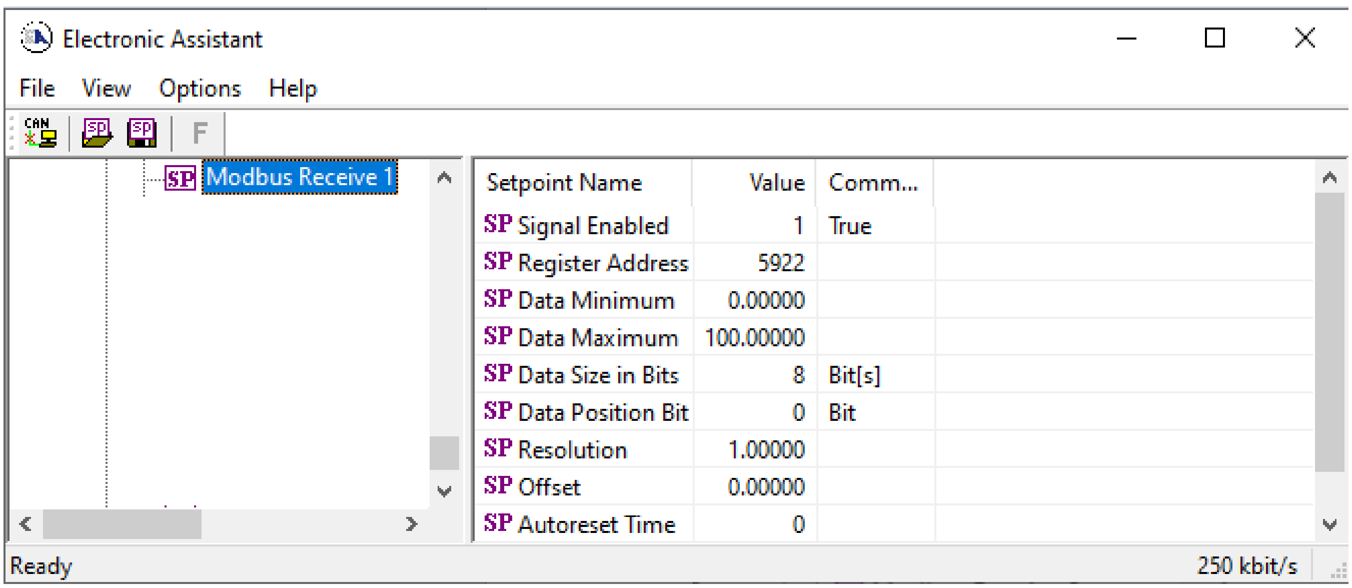


Figure 29 – The Screen Capture of Modbus Receive Function Block

Table 52 Modbus Function Block Setpoints

Name	**Range**	**Default**	**Notes**
Signal Enabled	Drop List	False	
Register Address	-		Check Modbus Address Map
Data Minimum	-65535 – 65535	0	
Data Maximum	-65535 – 65535	100	
Data Size in Bits	0 - 16	8	
Data Position Bit	0-15	0	
Resolution	0.001 – 0xFFFFFFFF	1	
Offset	-65535 – 65535	0	
Autoreset Time	0 – 60000	0	

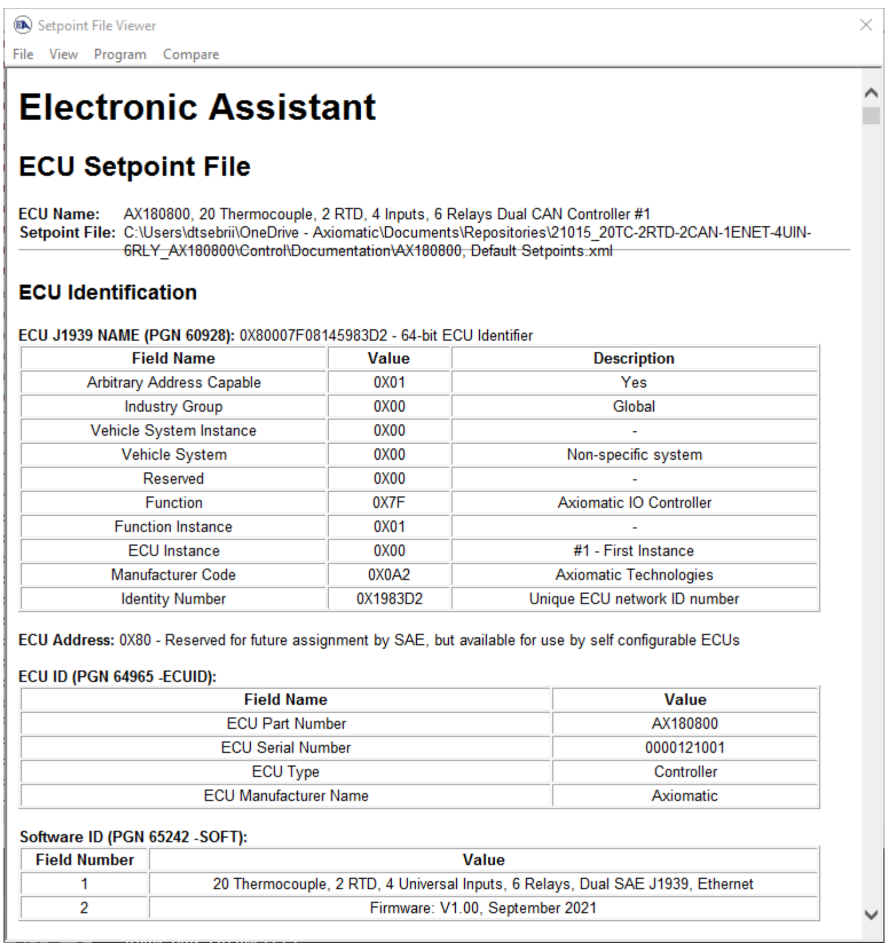
4.4 *Setpoint File*

The EA can store all converter configuration parameters in one setpoint file and then flash them into the controller in one operation.

The setpoint file is created and stored on disk using a command *Save Setpoint File* from the EA menu or toolbar. The user then can open the setpoint file, view or print it, and flash the setpoint file into the controller, see Figure 30.

The CAN network identification and “read-only” configuration parameters are not transferrable using this operation. Also, the controller will perform one or several internal resets of all function blocks during the setpoint flashing operation.

There can be small differences in configuration parameters between different versions of the application firmware. It is recommended that the user manually inspect all configuration parameters after flashing if the setpoint file was created by a different version of the application firmware.


Setpoint File Viewer

File View Program Compare

Electronic Assistant

ECU Setpoint File

ECU Name: AX180800, 20 Thermocouple, 2 RTD, 4 Inputs, 6 Relays Dual CAN Controller #1
Setpoint File: C:\Users\dtsebrii\OneDrive - Axiomatic\Documents\Repositories\21015_20TC-2RTD-2CAN-1ENET-4UIN-6RLY_AX180800\Control\Documentation\AX180800, Default Setpoints.xml

ECU Identification

ECU J1939 NAME (PGN 60928): 0X80007F08145983D2 - 64-bit ECU Identifier

Field Name	Value	Description
Arbitrary Address Capable	0X01	Yes
Industry Group	0X00	Global
Vehicle System Instance	0X00	-
Vehicle System	0X00	Non-specific system
Reserved	0X00	-
Function	0X7F	Axiomatic IO Controller
Function Instance	0X01	-
ECU Instance	0X00	#1 - First Instance
Manufacturer Code	0X0A2	Axiomatic Technologies
Identity Number	0X1983D2	Unique ECU network ID number

ECU Address: 0X80 - Reserved for future assignment by SAE, but available for use by self configurable ECUs

ECU ID (PGN 64965 -ECUID):

Field Name	Value
ECU Part Number	AX180800
ECU Serial Number	0000121001
ECU Type	Controller
ECU Manufacturer Name	Axiomatic

Software ID (PGN 65242 -SOFT):

Field Number	Value
1	20 Thermocouple, 2 RTD, 4 Universal Inputs, 6 Relays, Dual SAE J1939, Ethernet
2	Firmware: V1.00, September 2021

Figure 30. EA Setpoint File

A setpoint file containing default configuration parameters is available upon request.

5 FLASHING NEW FIRMWARE

When the new firmware becomes available, the user can replace the inclinometer 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 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 31.

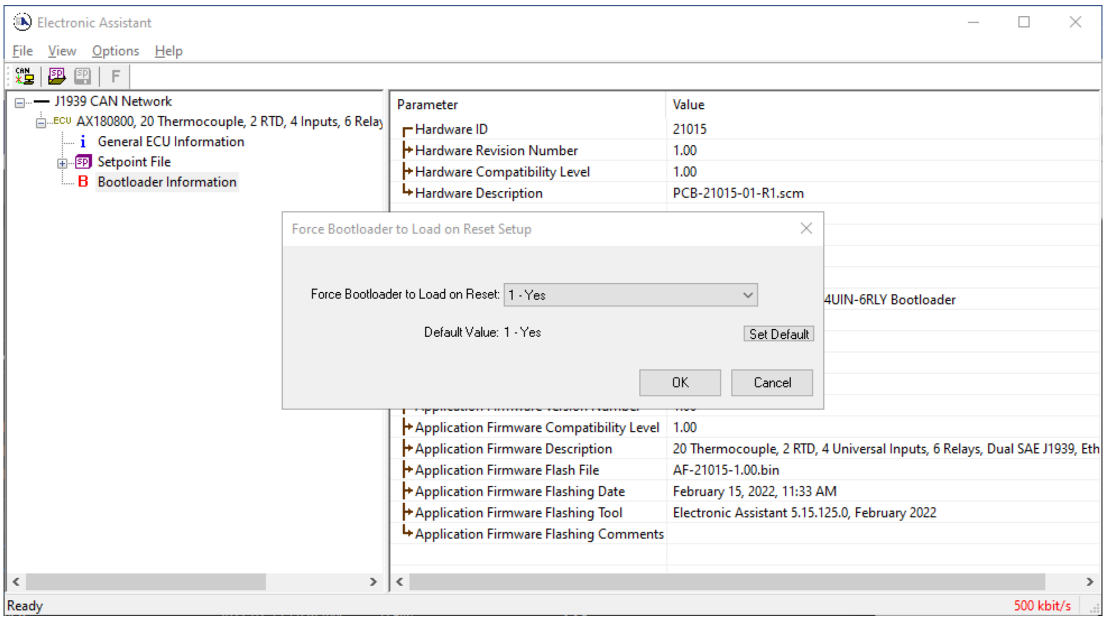


Figure 31. Bootloader Activation. First Step

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 required, see Figure 32. Select "Yes".

After automatic reset, instead of 20 Thermocouple, 2 RTD, 4 Inputs, 6 Relays Dual CAN Controller, *Ethernet*, 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.

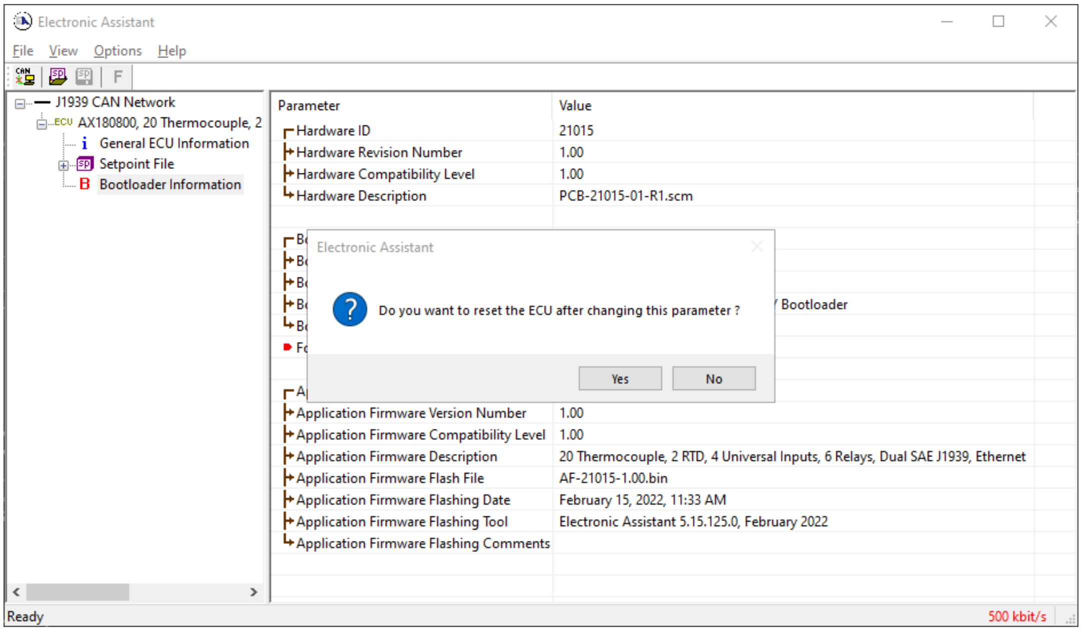


Figure 32. Bootloader Activation. Final Reset

All the bootloader specific information: controller 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 33. The information can be slightly different for different versions of the bootloader.

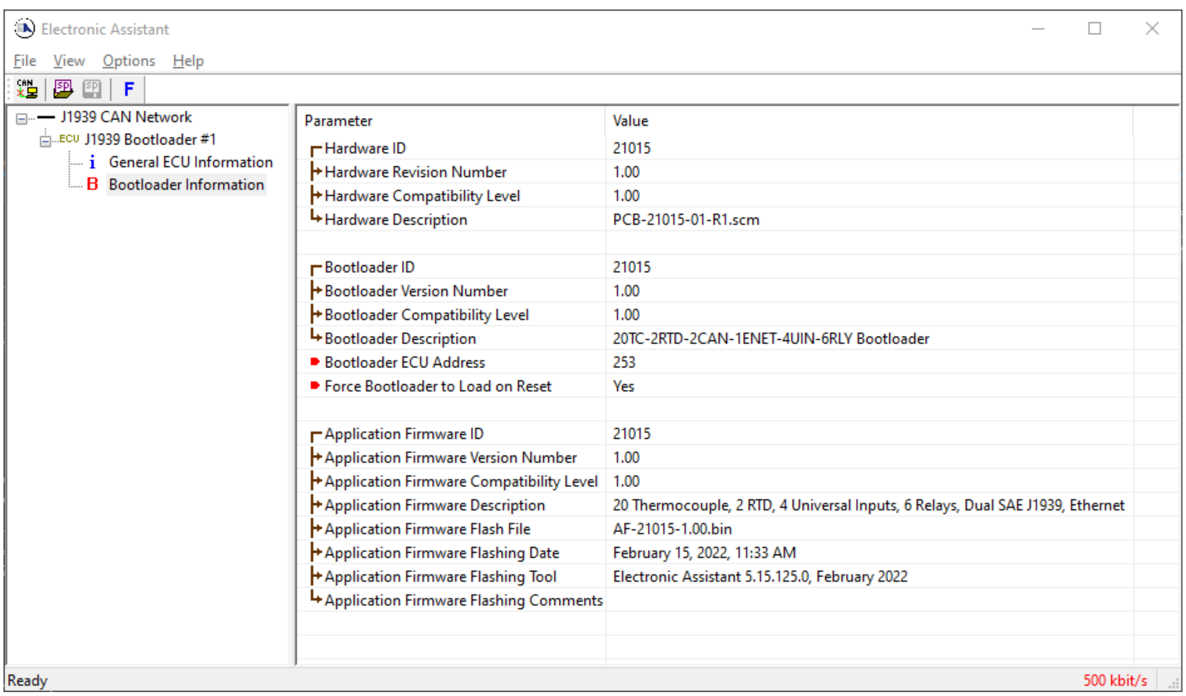


Figure 33. Bootloader Information Screen

At this point, the user can return to the installed controller 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 34.

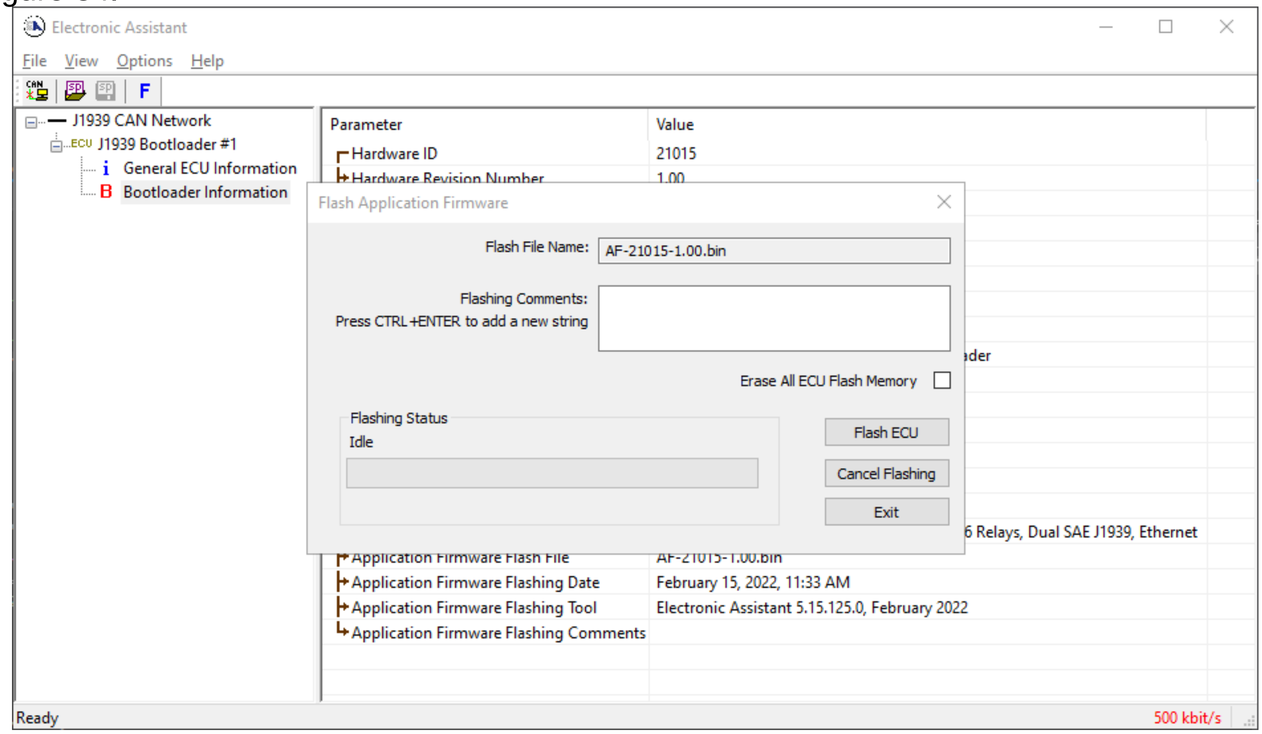


Figure 34. Flashing New Firmware. Preparation

[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 inclinometer flash memory. This operation, used in other products to reset configuration parameters kept in the flash memory to their default values, has no effect on this product. This is because the configuration parameters of the inclinometer are stored in a separate EEPROM memory.

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 35.

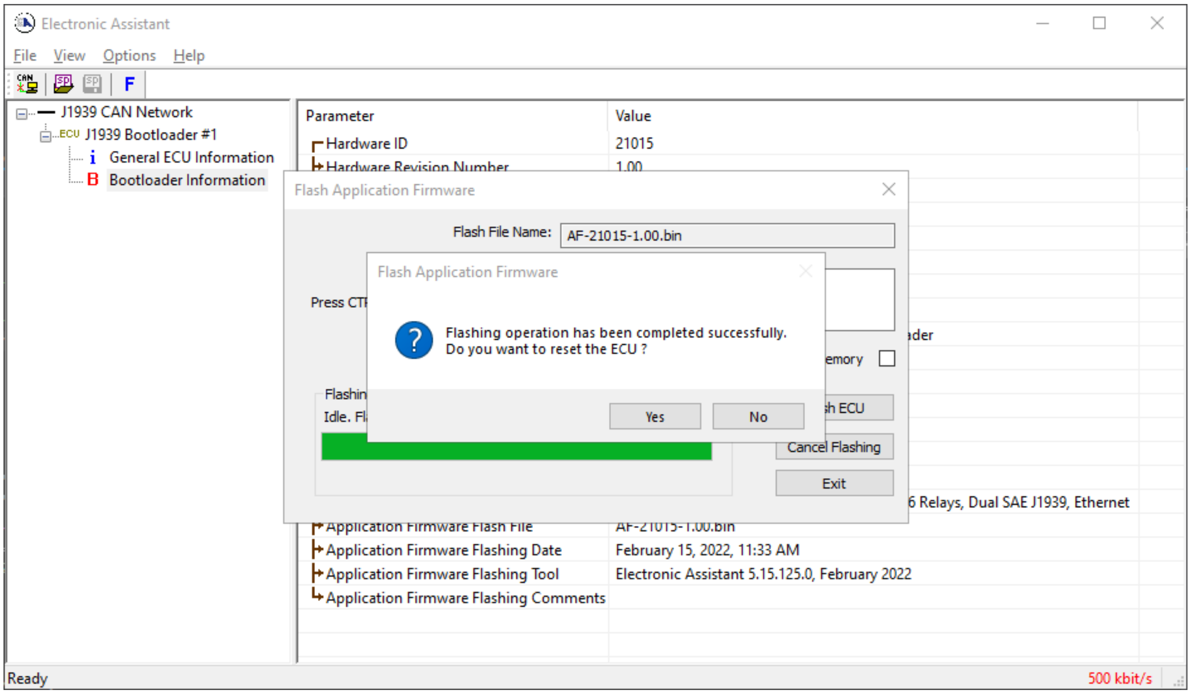


Figure 35. Flashing New Firmware. Final Reset.

Select *Yes* and see the ECU running the new firmware, see Figure 36. This will indicate that the flashing operation has been performed successfully.

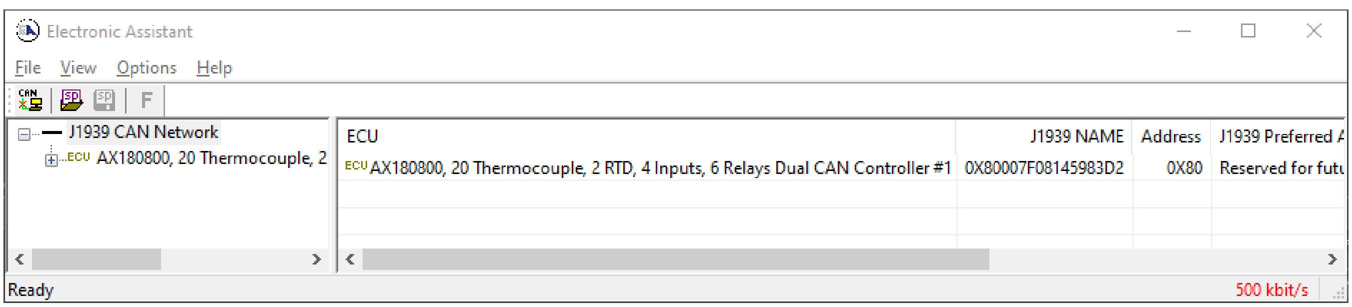


Figure 36. Firmware has been Updated. New Firmware Screen

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

6 TECHNICAL SPECIFICATIONS

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

6.1 Inputs

Power Supply Input	12V or 24VDC nominal (9…60 VDC power supply range)
Supply Current	158 mA at 12 V Typical 88 mA at 24 V Typical
Protection	Reverse polarity protection is provided. Power supply input section protects against transient surges and short circuits.
Isolation	Digital isolation is 400VDC from input to ground. Three-way isolation is provided for the CAN line, inputs and power supply.
Protection	Open circuit detection Over or under temperature detection High temperature shutdown detection
Thermocouple Types	Up to 20 channels, independently configurable for B, E, J, K, N, R, S or T
Thermocouple Inputs (20)	The device reads mV signals from the supported Thermocouples. B = 0 to 13.82 mV E = -9.835 to 76.373 mV J = -8.095 to 69.553 mV K = -6.458 to 54.886 mV N = -4.345 to 47.513 mV R = -0.226 to 21.101 mV S = -0.236 to 18.693 mV T = -6.258 to 20.872 mV Temperatures are configured to indicate the SAE J1939 SPN to be transmitted by that temperature input. Accuracy: +/- 1°C typical with cold junction compensation at ambient temperature Resolution: 0.001°C
TC Scan Rate	100ms per channel, total sweep time maximum 2.2 seconds
Common Mode Readings	Input range +/- 2.5V maximum Rejection is 120db (maximum) at 2.5Vp-p (50-60Hz)
Thermal Drift	4 ppm/°C of span (maximum)
RTD Types	Up to 2 channels for Rt100 sensor type inputs with 2 or 3-wire connection. Each channel operates independently.
RTD Inputs (2)	The device accepts inputs within the following range of 20 - 400 Ohms. RTD lead resistance range is 0 – 10 Ohms. Accuracy: +/- 0.4°C with offset calibration performed at R_{RTD} = 100R (typical at ambient temperature) Resolution: 0.001°C
RTD Scan Rate	20 samples per second for all 2 channels

Signal Inputs (4)	4 Universal Signal Inputs configurable as: Voltage, Current, Frequency, PWM or Digital 12-bit Analog to Digital resolution (Voltage and Current) Protected against shorts to GND or +Vps Voltage Types: 0-5Vdc (Input impedance 204 kΩ) or 0-10Vdc (Input impedance 134 kΩ) 1mV resolution, accuracy +/- 0.2% error Current Types: 0-20 mA (Input impedance 249 kΩ) or 4-20 mA (Input impedance 249 kΩ) 1uA resolution, accuracy +/- 0.2% error Frequency Type: Range: 0.5 Hz to 10 kHz 0.01% resolution, accuracy +/- 0.1% error PWM Type: PWM Signal Frequency: 1 Hz to 10 kHz PWM Duty Cycle: 0 to 100% 0.01% resolution, accuracy +/- 1% error Digital Type: Active High with 10K Pullup or Active Low with 10K Pulldown resistor to GND Amplitude: up to +Vsupply

6.2 Outputs

Relay Outputs (6)	6 Form C relay outputs 5 A at 30 VDC NO, COM and NC pins are available.

6.3 Communication

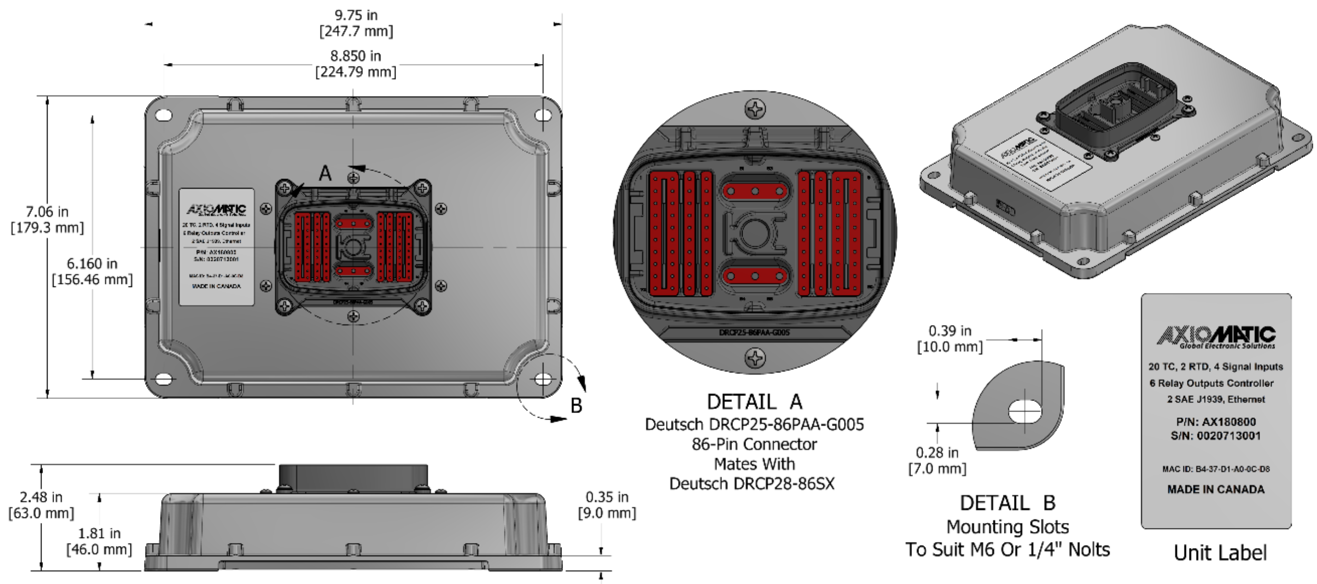
CAN	2 CAN 2.0B port, protocol SAE J1939 Auto-Baud-Rate-Detection 400VDC isolation The two CAN channels are fully isolated from one another in the hardware which allows for different CAN networks to be connected. In the software, each will also act independently where incoming or outgoing messages can be configured to receive or send on a specific CAN channel.
Network Termination	According to the CAN standard, it is necessary to terminate the network with external termination resistors. The resistors are 120 Ohm, 0.25W minimum, metal film or similar type. They should be placed between CAN_H and CAN_L terminals at both ends of the network.
Ethernet	1 port 10/100 Mbit Ethernet compliant 10BASE-T. 100BASE-Tx (Auto-configuration and full-duplex is supported.) Auto-MDIX Modbus TCP/IP

6.4 General Specifications

Microcontroller	STM32F407ZGT6, 1 Mbyte Flash Memory, 192+4 Kbyte SRAM
Control Logic	User programmable functionality with the Electronic Assistant Refer to the User Manual.
User Interface	Electronic Assistant, P/N: AX070502 or AX070506K Updates for the EA are found on www.axiomatic.com under the log-in tab.
Vibration	Pending
Shock	Pending
Operating Temperature Range	-40 to 85 ºC (-40 to 185 ºF)
Storage Temperature Range	-50 to 120 ºC (-58 to 248 ºF)
Humidity	Protected against 95% humidity non-condensing, 30 ºC to 60 ºC
Protection	IP67
Weight	3.1 lb. (0.141 kg)

Enclosure	Rugged aluminum enclosure, anodized, gasket Connector: 86-pin TE Deutsch equivalent DRCP25-86PAA-G005s Mates with DRCP28-86SX Notes: SECURE HARNESS WITH TIE WRAPS FOR HIGH VIBRATION APPLICATIONS. REQUIRES COVER TE DEUTSCH equivalent PN 4828-008-8605 (NOT INCLUDED) FOR HIGH PRESSURE SPRAY APPLICATIONS.

6.5 Dimensional Drawing



CONNECTOR PINOUT TABLE				
Pin1: TC2_IN-	Pin19: TC5_IN-	Pin37: COM_6	Pin55: RTD_B_2	Pin73: TC20_IN-
Pin2: TC2_IN+	Pin20: TC5_IN+	Pin38: NO_6	Pin56: RTD_T_1	Pin74: TC20_IN+
Pin3: TC1_IN-	Pin21: NO_3	Pin39: Spare	Pin57: TC16_IN-	Pin75: TC19_IN-
Pin4: TC1_IN+	Pin22: NC_4	Pin40: Spare	Pin58: TC16_IN+	Pin76: TC19_IN+
Pin5: NC_1	Pin23: COM_4	Pin41: TC12_IN-	Pin59: TC15_IN-	Pin77: ETH_RX_X
Pin6: COM_1	Pin24: NO_4	Pin42: TC12_IN+	Pin60: TC15_IN+	Pin78: ETH_RX_P
Pin7: NO_1	Pin25: TC8_IN-	Pin43: TC11_IN-	Pin61: RTD_M_1	Pin79: ETH_TX_N
Pin8: NC_2	Pin26: TC8_IN+	Pin44: TC11_IN+	Pin62: RTD_B_1	Pin80: ETH_TX_P
Pin9: TC4_IN-	Pin27: TC7_IN-	Pin45: IN1	Pin63: GND	Pin81: BATT-
Pin10: TC4_IN+	Pin28: TC7_IN+	Pin46: IN2	Pin64: GND	Pin82: BATT+
Pin11: TC3_IN-	Pin29: NC_5	Pin47: IN3	Pin65: TC18_IN-	Pin83: Spare
Pin12: TC3_IN+	Pin30: COM_5	Pin48: IN4	Pin66: TC18_IN+	Pin84: BATT-
Pin13: COM_2	Pin31: NO_5	Pin49: TC14_IN-	Pin67: TC17_IN-	Pin85: BATT+
Pin14: NO_2	Pin32: NC_6	Pin50: TC14_IN+	Pin68: TC17_IN+	Pin86: Spare
Pin15: NC_3	Pin33: TC10_IN-	Pin51: TC13_IN-	Pin69: CAN2_P	
Pin16: COM_3	Pin34: TC10_IN+	Pin52: TC13_IN+	Pin70: CAN2_N	
Pin17: TC6_IN-	Pin35: TC9_IN-	Pin53: RTD_T_2	Pin71: CAN_P	
Pin18: TC6_IN+	Pin36: TC9_IN+	Pin54: RTD_M_2	Pin72: CAN_N	

7 THIRD-PARTY SOFTWARE LICENSE NOTICES

This section contains Third-Party Software License Notices and/or Additional Terms and Conditions for licensed third-party software components included in the 20 Thermocouple, 2 RTD, 4 Inputs, 6 Relays Dual CAN Controller and Ethernet firmware.

Table 53. Third-Party Software License Notices

Third-Party Software	License Notice/Terms
STMicroelectronics microcontroller support software	COPYRIGHT(c) 2017 STMicroelectronics Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. Licensed under MCD-ST Liberty SW License Agreement V2, (the "License"); You may not use this file except in compliance with the License. You may obtain a copy of the License at: http://www.st.com/software_license_agreement_liberty_v2 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
FreeRTOS Kernel V10.2.0	Copyright (C) 2019 Amazon.com, Inc. or its affiliates. All Rights Reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Third-Party Software	License Notice/Terms
	THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
FreeRTOS+TCP V2.2.0	Copyright (C) 2017 Amazon.com, Inc. or its affiliates. All Rights Reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

8 VERSION HISTORY

User Manual Version	Firmware version	Electronic Assistant (EA) version	Date	Author	Modifications
1	1.xx	5.15.125.0	April 8, 2022	Dmytro Tsebrii	• Initial Release
1A	1.xx	5.15.125.0	December7, 2022	Dmytro Tsebrii	• Fixed Modbus section. • Removed unused references
2	2.xx		November 2, 2023		• Added Modbus Receive function block section • Modified information for CAN Rx and CAN Tx
2A	2.xx		December 31, 2023	M Ejaz, Sue Thomas	• Marketing review, legacy updates, new address

Global Electronic Solutions

OUR PRODUCTS

AC/DC Power Supplies

Actuator Controls/Interfaces

Automotive Ethernet Interfaces

Battery Chargers

CAN Controls, Routers, Repeaters

CAN/WiFi, CAN/Bluetooth, Routers

Current/Voltage/PWM Converters

DC/DC Power Converters

Engine Temperature Scanners

Ethernet/CAN Converters, Gateways, Switches

Fan Drive Controllers

Gateways, CAN/Modbus, RS-232

Gyroscopes, Inclinometers

Hydraulic Valve Controllers

Inclinometers, Triaxial

I/O Controls

LVDT Signal Converters

Machine Controls

Modbus, RS-422, RS-485 Controls

Motor Controls, Inverters

Power Supplies, DC/DC, AC/DC

PWM Signal Converters/Isolators

Resolver Signal Conditioners

Service Tools

Signal Conditioners, Converters

Strain Gauge CAN Controls

Surge Suppressors

OUR COMPANY

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

QUALITY DESIGN AND MANUFACTURING

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

WARRANTY, APPLICATION APPROVALS/LIMITATIONS

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

COMPLIANCE

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

SAFE USE

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

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

SERVICE

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

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

DISPOSAL

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

CONTACTS

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

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