RS232-RS232-RS422 ROUTER WITH ETHERNET AND CAN

USER MANUAL

P/N: AX142100

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

VERSION HISTORY

Version	Date	Author	Modification
1.0.0.	Oct. 1, 2020	Antti Keränen	Initial Draft
1.0.1.	Nov. 12, 2020	Antti Keränen	New pinout description, datasheet section updated.
1.0.2.	Nov. 18, 2020	Antti Keränen	Serial port pinouts corrected in section 4.1
1.0.3.	Feb. 23, 2021	Antti Keränen	Web server configuration options description updated. Example TCP client C code added. Ethernet frame contents description enhanced. Description of cURL settings upload/download added.
1.0.4	Nov. 23, 2023	M Ejaz	Marketing review Updated technical specifications

ACCRONYMS

ACK	Positive Acknowledgement (from SAE J1939 standard)
BATT +/-	Battery positive (a.k.a. Vps) or Battery Negative (a.k.a. GND)
DM	Diagnostic Message (from SAE J1939 standard)
DTC	Diagnostic Trouble Code (from SAE J1939 standard)
EA	Axiomatic Electronic Assistant (A Service Tool for Axiomatic ECUs)
ECU	Electronic Control Unit (from SAE J1939 standard)
GND	Ground reference (a.k.a. BATT-)
I/O	Inputs and Outputs
IP	Internet Protocol
MAC	Media Access Control
MAP	Memory Access Protocol
NAK	Negative Acknowledgement (from SAE J1939 standard)
PDU1	A format for messages that are to be sent to a destination address, either specific or global (from SAE J1939 standard)
PDU2	A format used to send information that has been labeled using the Group Extension technique, and does not contain a destination address.
PGN	Parameter Group Number (from SAE J1939 standard)
PropA	Message that uses the Proprietary A PGN for peer-to-peer communication
PropB	Message that uses a Proprietary B PGN for broadcast communication
SPN	Suspect Parameter Number (from SAE J1939 standard)
TCP/IP	Transmission Control Protocol / Internet Protocol
TP	Transport Protocol
Vps	Voltage Power Supply (a.k.a. BATT+)

TABLE OF CONTENTS

List of Figures

List of Tables

REFERENCES

J1939	Recommended Practice for a Serial Control and Communications Vehicle Network, SAE, April 2011
J1939/21	Data Link Layer, SAE, December 2010
J1939/71	Vehicle Application Layer, SAE, March 2011
J1939/73	Application Layer-Diagnostics, SAE, February 2010
J1939/81	Network Management, SAE, March 2017
TDAX142100	Technical Datasheet, RS232-RS232-RS422-ENET-CAN Converter, Axiomatic Technologies 2023
UMAX07050x	User Manual, Axiomatic Electronic Assistant and USB-CAN, Axiomatic Technologies, June 2023

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

NOTE: This product is supported by Axiomatic Electronic Assistant

1. OVERVIEW OF CONTROLLER

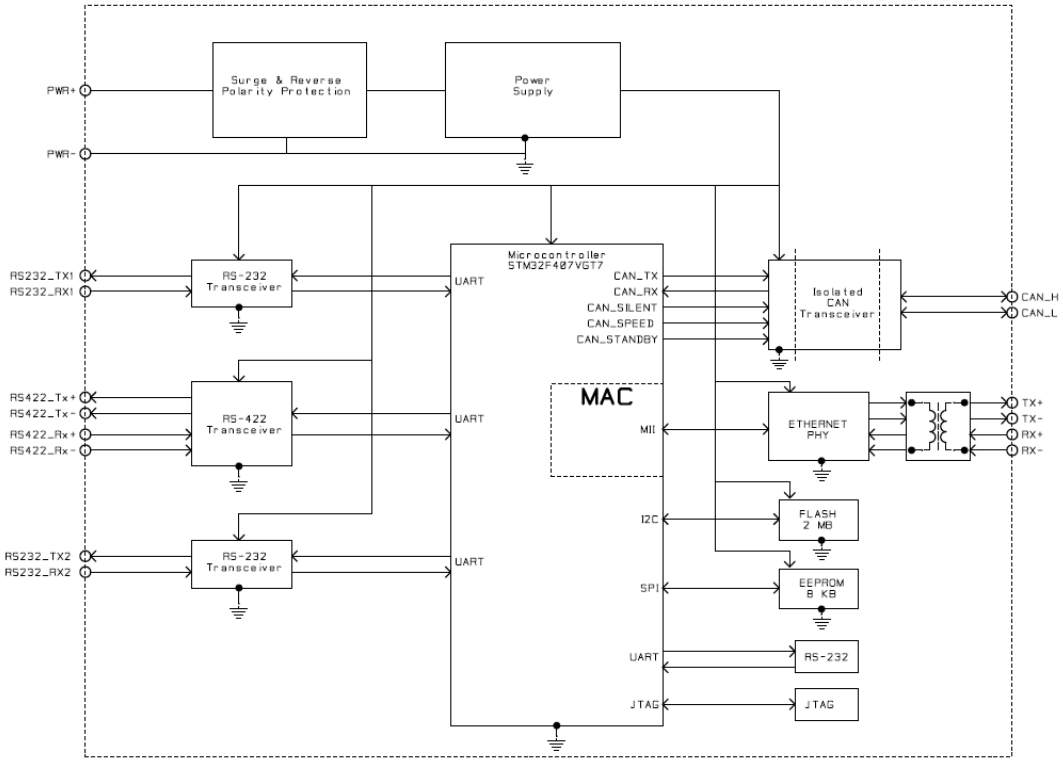


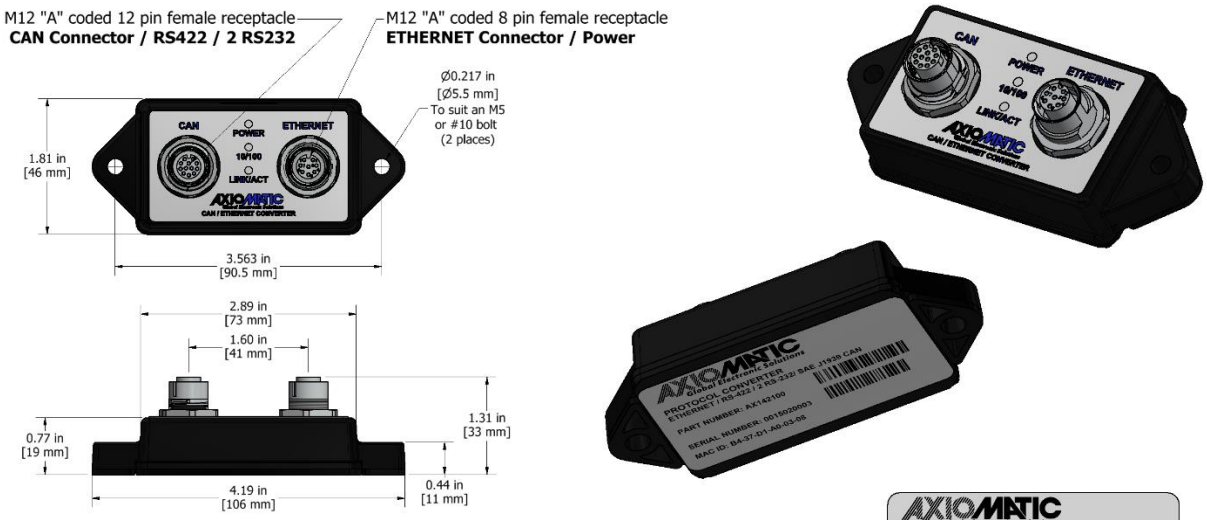
Figure 1 – Block diagram of the RS232-RS232-RS422 Router with Ethernet and CAN

The RS232-RS232-RS422 Router with Ethernet and CAN (later 3RS-ENET-CAN) electronic control unit (ECU) is a device that forwards serial port messages between the three serial ports, CAN and Ethernet based on a custom routing configuration. The configuration can be done using a web browser and the built-in web server running on the 3RS-ENET-CAN device.

The Axiomatic Electronic Assistant is used to configure the network parameters of the 3RS-ENET-CAN ECU. The configuration of the rest of the parameters can be done via the web browser interface (port 80).

2. INSTALLATION INSTRUCTIONS

2.1. Dimensions and Pinout



M12 12 PIN FEMALE CONNECTOR			
PIN	FUNCTION	PIN	FUNCTION
1	RS422_RX+	7	CAN_H
2	RS422_TX+	8	RS232_TX1
3	RS422_RX-	9	RS232_RX1
4	RS232_TX2	10	RS422_TX-
5	RS232_RX2	11	GND
6	CAN_L	12	GND

M12 8 PIN FEMALE CONNECTOR			
PIN	FUNCTION	PIN	FUNCTION
1	PWR+	5	ETH_RX+
2	PWR-	6	ETH_TX+
3	PWR-	7	PWR+
4	ETH_TX-	8	ETH_RX-

Figure 2 – Controller Dimensions and Label

CAN connector / 2xRS232		Ethernet connector / RS422	
Pin #	**Function**	**Pin #**	**Function**
1	RS422 RX +	1	Power +
2	RS422 TX +	2	Power -
3	RS422 RX -	3	Power -
4	RS232, #2 TX	4	Ethernet TX -
5	RS232, #2 RX	5	Ethernet RX +
6	CAN L	6	Ethernet TX +
7	CAN H	7	Power +
8	RS232 #1, TX	8	Ethernet RX -
9	RS232 #1, RX		
10	RS422 TX -		
11	GND		
12	GND		

Table 1 – AX142100 Connector Pinout

3. OVERVIEW OF J1939 FEATURES

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

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

3.1. Introduction to Supported Messages

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

From J1939-21 – Data Link Layer

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

From J1939-73 – Diagnostics

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

From J1939-81 – Network Management

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

From J1939-71 – Vehicle Application Layer

• ECU Identification Information	64965	0x00FDC5
• Software Identification	65242	0x00FEDA
• Component Identification	65259	0x00FEEB

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

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

3.2. NAME, Address and Identification Information

The 3RS-ENET-CAN ECU has the following default for the J1939 NAME. The user should refer to the SAE J1939/81 standard for more information on these parameters and their ranges.

Arbitrary Address Capable	Yes
Industry Group	0, Global
Vehicle System Instance	0
Vehicle System	0, Non-specific system
Function	25, Axiomatic Protocol Converter
Function Instance	21, Axiomatic AX142100
ECU Instance	0, First Instance
Manufacture Code	162, Axiomatic Technologies
Identity Number	Variable, uniquely assigned during factory programming for each ECU

The ECU Instance is a configurable setpoint associated with the NAME. Changing this value will allow multiple ECUs of this type to be distinguishable from one another when they are connected on the same network.

The default value of the “ECU Address” setpoint is 128 (0x80), which is the preferred starting address for self-configurable ECUs as set by the SAE in J1939 tables B3 and B7. The EA will allow the selection of any address between 0 and 253. ***It is user’s responsibility to select an address that complies with the standard***. The user must also be aware that since the unit is arbitrary address capable, if another ECU with a higher priority NAME contends for the selected address, the 10 Analog input will continue select the next highest address until it finds one that it can claim. See J1939/81 for more details about address claiming.

ECU Identification Information

PGN 64965		ECU Identification Information	-ECUID
Transmission Repetition Rate:		On request	
Data Length:		Variable	
Extended Data Page:		0	
Data Page:		0	
PDU Format:		253	
PDU Specific:		197 PGN Supporting Information:	
Default Priority:		6	
Parameter Group Number:		64965 (0x00FDC5)	
Start Position	Length	Parameter Name	SPN
a	Variable	ECU Part Number, Delimiter (ASCII “*”)	2901
b	Variable	ECU Serial Number, Delimiter (ASCII “*”)	2902
c	Variable	ECU Location, Delimiter (ASCII “*”)	2903
d	Variable	ECU Type, Delimiter (ASCII “*”)	2904
e	Variable	ECU Manufacturer Name, Delimiter (ASCII “*”)	4304
(a)*(b)*(c)*(d)*(e)*			

Software Identifier

PGN 65242		Software Identification	-SOFT
Transmission Repetition Rate:		On request	
Data Length:		Variable	
Extended Data Page:		0	
Data Page:		0	
PDU Format:		254	
PDU Specific:		218 PGN Supporting Information:	
Default Priority:		6	
Parameter Group Number:		65242 (0x00FEDA)	
Start Position	Length	Parameter Name	SPN
1	1 Byte	Number of software identification fields	965
2-n	Variable	Software identification(s), Delimiter (ASCII “*”)	234

Byte 1 is set to 5, and the identification fields are as follows.

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

The EA shows all this information in its “General ECU Information” page. *Note: The information provided in the Software ID is available for any J1939 service tool which supports the PGN -SOFT*

Component Identification

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

4. WEB BROWSER BASED CONTROLLER CONFIGURATION

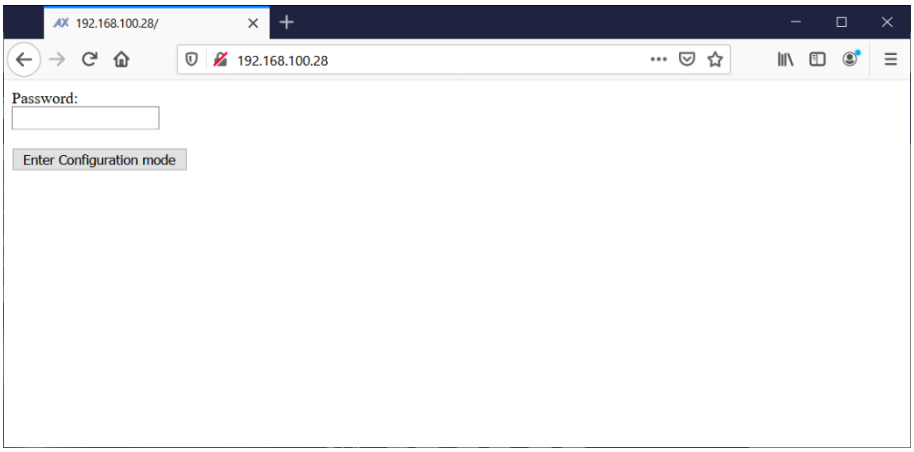
The 3RS-ENET-CAN controller supports configuration of the data routing parameters from Ethernet port using a standard web browser.

4.1. Parameter Editing

The 3RS-ENET-CAN has a web server running on TCP port 80. This web server has the following pages implemented:

<configured ip>/config

The configuration page asks for a password. The default password is '**AX142100**' (this is case sensitive).



When the correct password is entered, the configuration page is opened. The settings can be applied by clicking the button at the bottom of the page. In case the user doesn't want to change settings, the connection can be closed.

The configuration page allows the user to modify the device's IP address, netmask and the main configuration parameters for the communication interfaces. The CAN configuration parameters include the default baud rate to use and the auto-baud rate capability.

The serial port configuration contains, baud rate (freely settable), number of data, start and stop bits and parity.

The serial port configuration also supports custom message delimiter character. By default, only the detected idle condition on the serial interface is considered as a message delimiter. By configuring a customer message delimiter character, messages can be picked up from a continuous serial data stream.

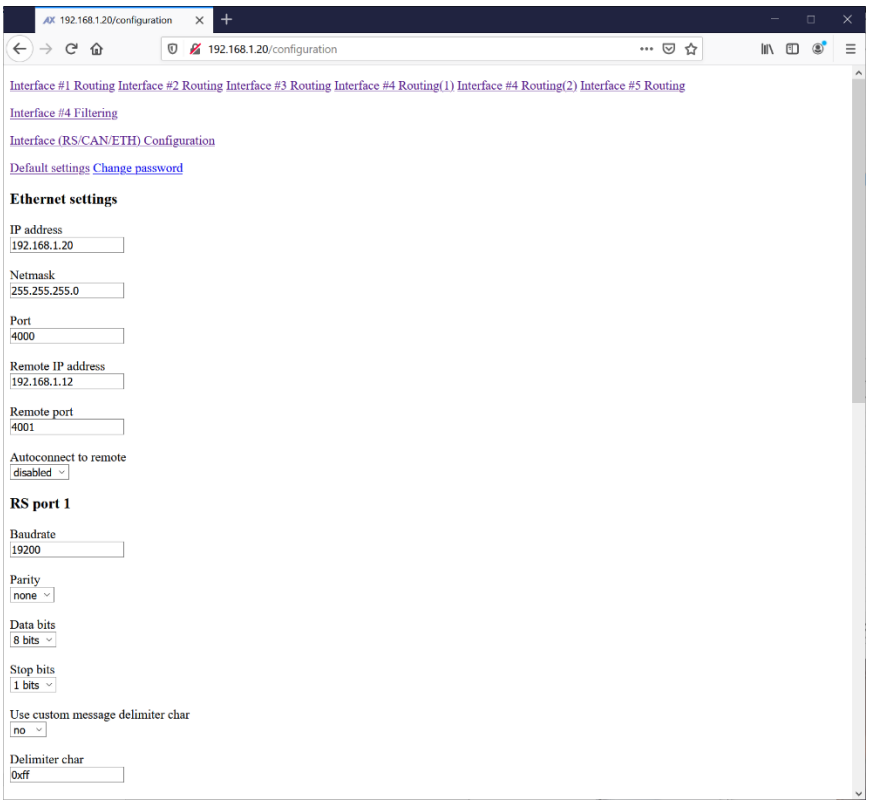
In the settings (see also

Table 1 – AX142100 Connector Pinout)

RS Port 1 == RS232, pins 8 & 9 of the CAN / RS232 / RS422 Connector

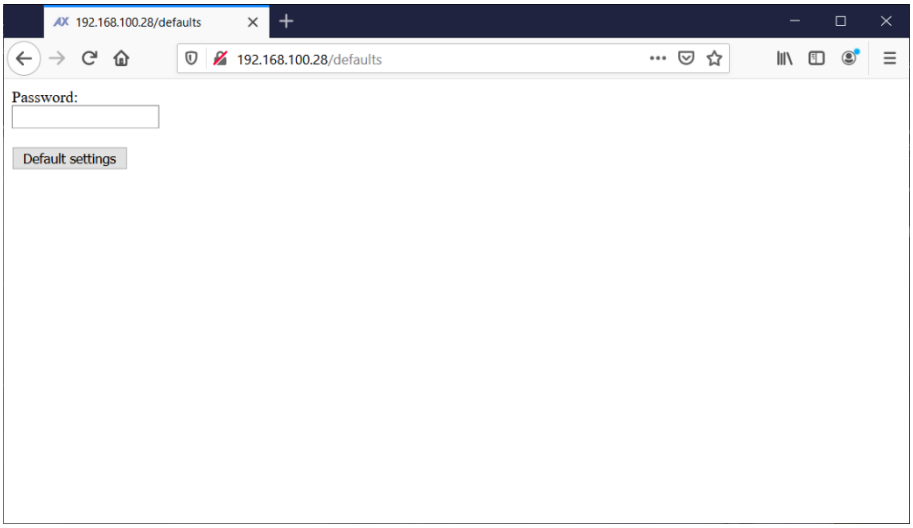
RS Port 2 == RS232, pins 4 & 5 of the CAN / RS232 / RS422 Connector

RS Port 3 == RS422, pins 1, 2, 3 & 10 of the CAN / RS232 / RS422 Connector



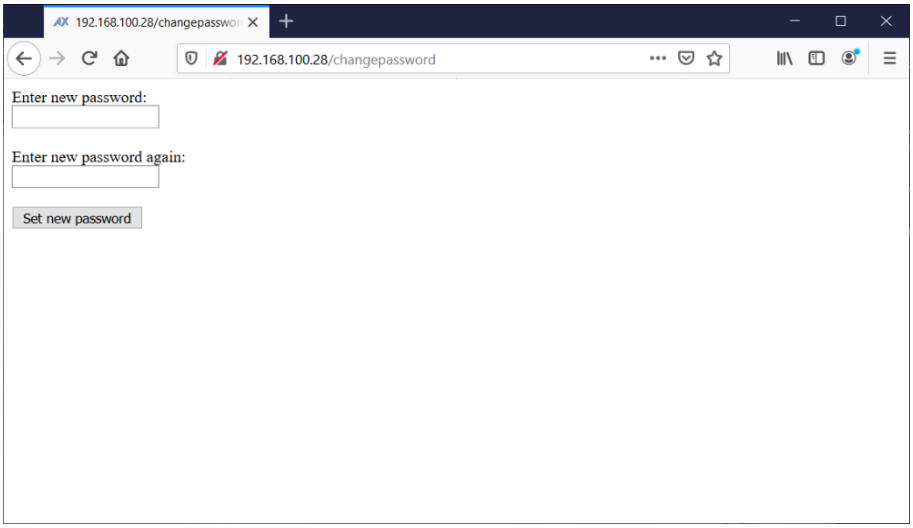
<configured ip>/defaults

The controller can be reverted to default settings using the ‘/defaults’ page. The default password is ‘**AX142100**’ (also case sensitive).



<configured ip>/changepassword

The configuration password can be changed using the ‘/changepassword’ page. The default password is ‘**AX142100**’ (also case sensitive).



<configured ip>/routing1…3, 5

The data routing configuration is done for each interface separately. The routing is done for all frames received from the serial ports and from Ethernet TCP connection.

Each of the rules have a list of Output Interfaces, Match Bytes and Mask Bytes (software filter), add Start and End Bytes and CAN options, such as add CAN ID, use 29/11 bit ID and whether to use TP or not. For data forwarded to CAN interface, it is also possible to use CAN message byte 0 as an index.

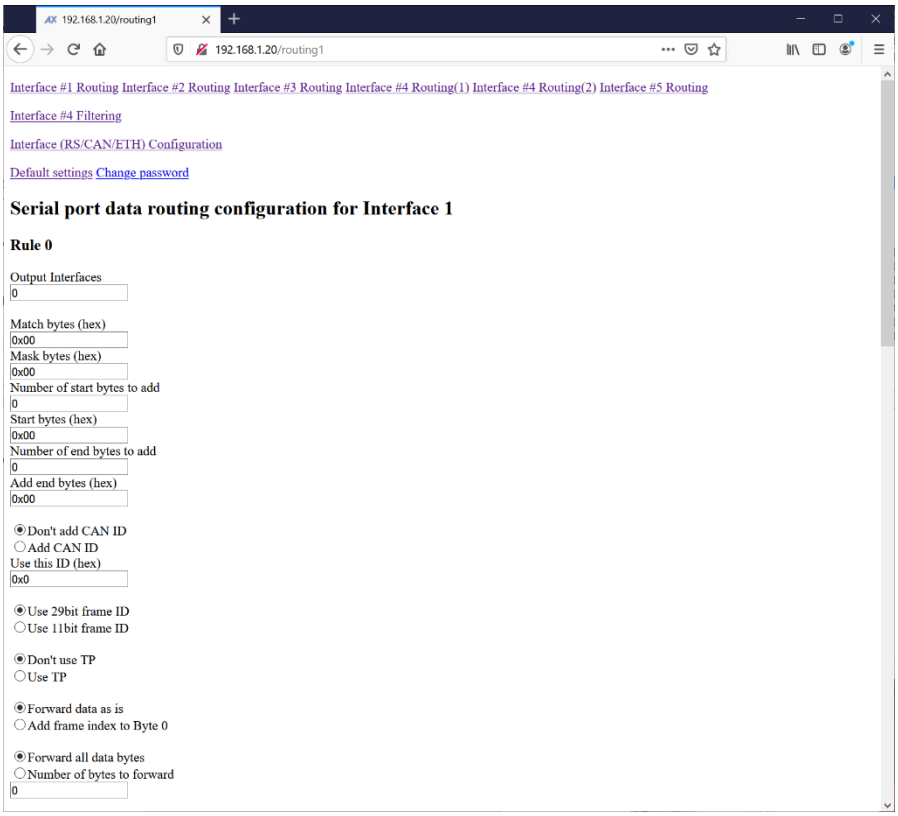
The Output Interfaces should be entered as comma separated list with no spaces. Match and Mask Bytes define a software filter for selecting the frames that will be routed to the configured Output Interfaces.

Start bytes and Number of start bytes to add define the bytes that should be added to the beginning of the forwarded frame.

End Bytes and Number of end bytes to add define the bytes that should be added to the end of the forwarded frame.

In case the CAN Interface (interface #4) is among the Output interfaces, the forwarded frames that end up to CAN bus can be configured to have a specific CAN frame ID. In case a CAN frame ID is not defined, the first 29/11 bits (depending on the CAN ID type) will be used as the CAN frame ID.

In case Use TP is selected, the forwarded frame will be wrapped to TP frames in case the length exceeds 8 bytes. In case TP is not used, the frame will be sent as multiple single CAN frames. The option to add frame index to byte 0 has an effect only if TP is not used.



The Match and Mask Bytes are applied like this on the received serial port data. In case comparison is true, the data is forwarded.

"RX data & mask" == match

The Match bytes (hex), Mask bytes (hex), Number of start bytes to add, Start bytes (hex), Number of end bytes to add and Add end bytes (hex) are applied to the received serial data.

Don’t add CAN ID/Add CAN ID, Use this ID (hex). Use 29bit frame ID/Use 11bit frame ID, Don’t use TP/Use TP and Forward data as is/Add frame index to byte 0 are applied to data that is forwarded to the CAN interface (#4).

The Forward all data bytes/Number of bytes to forward are applied to all forwarded data.

Please note, that the TP messaging is used only when 29bit CAN frame ID is specified. In case TP is in use, the PGN wrapped inside the TP frame is specified using the Use this ID (hex) option.

The Add frame index to Byte 0 option can be used with 11bit frame IDs. This implements “TP like” CAN output.

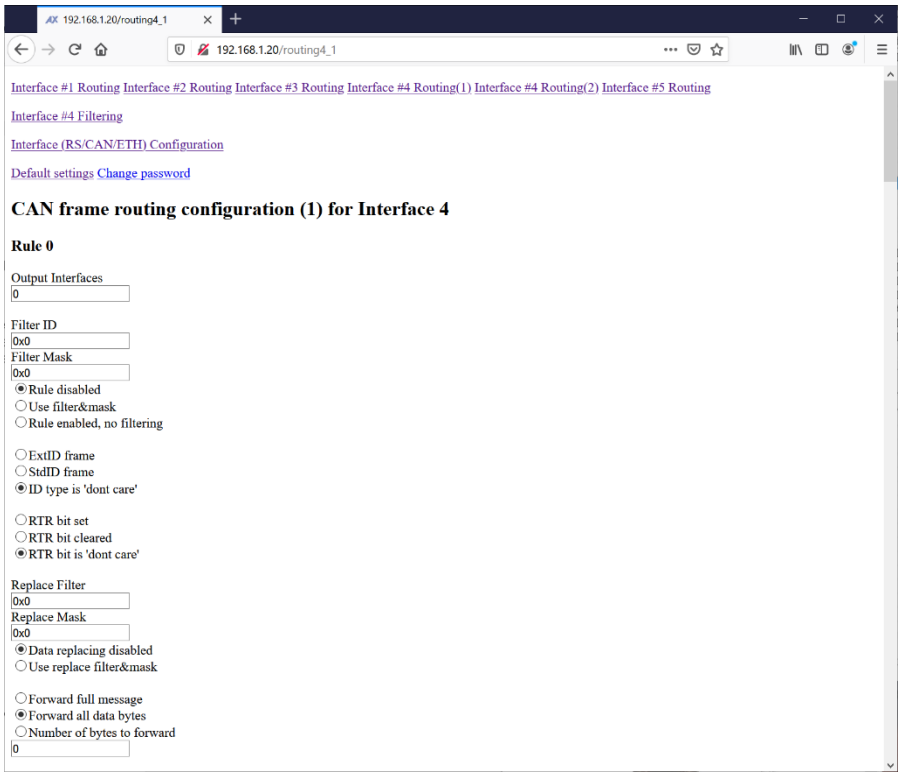
<configured ip>/routing4_1 & _2

CAN interface supports 16 data routing rules. The rules are configured using two pages, first one containing rules 0…7 and the second containing rules 8…15.

Each of the rules have a list of Output Interfaces, Filter ID and Mask (software filter) and Data replacing options. The data replacing is supported for the CAN Frame ID bits. Also the number of data bytes to forward can be specified.

The Output Interfaces should be entered as comma separated list with no spaces. Filter ID and Mask are identical to the hardware filter configuration, these two settings are used in a software filter for selecting the frames that will be routed to the configured Output Interfaces.

The Replace Filter and Mask can be used for example to modify the Source Address, PGN and/or Priority bits of the J1939 frame. The data replacing function is applied for all frames that pass the software filter and will be done before routing the frame to the configured output interfaces.



To forward all data from frames with a PGN 0xFF01 to interfaces 2 & 3 and modify the forwarded PGN to 0xFF82, the following setup would need to be used:

Output Interfaces: **2,3**
Filter ID: **0xFF0100**
Filter Mask: **0xFFFF00**
'**Use filter&mask**', '**ExtID frame**' and '**RTR bit is don't care**' selected
Replace Filter: **0xFF8200**
Replace Mask: **0xFFFF00**
'**Use replace filter&mask**' selected
'**Forward all data bytes**' selected

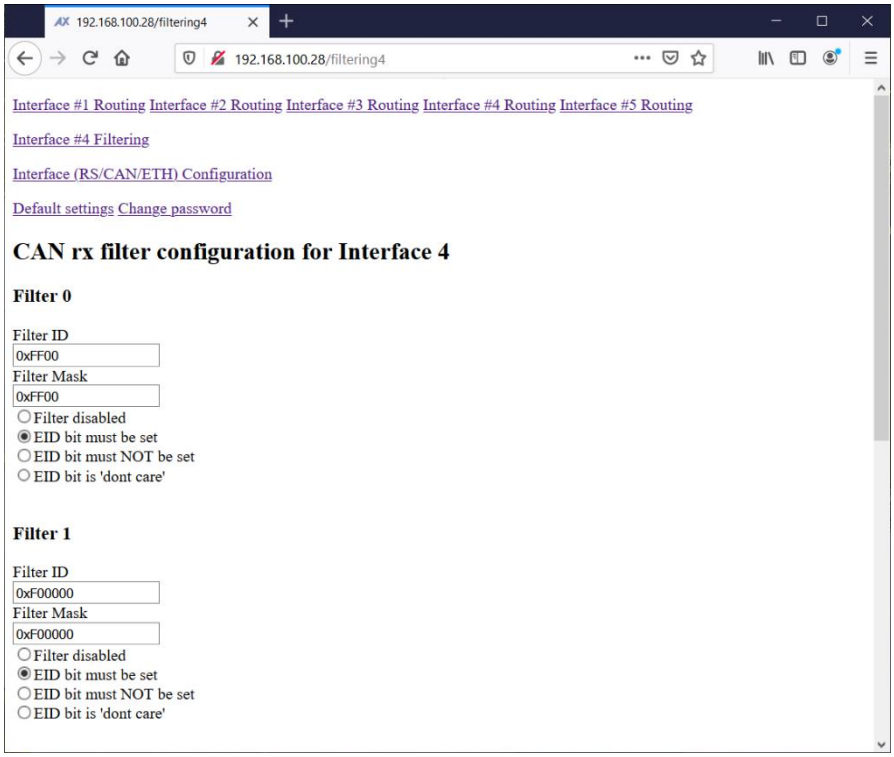
<configured ip>/filtering4

The receive filter is used for selecting which CAN frames will be received. All received CAN frames that pass the reception filter will be forwarded to the data routing module.

The configured CAN ID filter will be assigned to the CAN interface's acceptance filter registers. No additional software filtering will be done in the message reception. However, the data routing module supports software filtering for selecting the frames that will be routed.

In case all filters are disabled, no CAN frames will be accepted.

The CAN Filter ID defines the 29-bit extended or 11-bit standard frame ID. The Filter Mask bit '1' forces the compare, '0' marks the bit as 'don't care'. To configure a filter for receiving all possible frames, the ID and Mask should be set to '0' and option for EID bit as 'don't care'.



<configured ip>/fullconfig

The 3RS-ENET-CAN supports the use of cURL (or equivalent) for full settings file download and upload.

The current configuration can be downloaded to PC using command:

```
curl --http0.9 -o "./config.file" "http://192.168.100.28/fullconfig"
```

The saved configuration can be uploaded to the 6CAN-ENET device:

```
curl --http0.9 --upload-file "./config.file" "http://192.168.100.28/fullconfig"
```

The above examples were tested using Windows 10 and **curl-7.70.0-win64-mingw** available from https://curl.haxx.se/windows/

4.2. TCP Connections

The forwarded frames can be sent as proprietary TCP frames. A client can listen to these frames by initiating a TCP connection to port 4000 (or to custom port, configured using EA or a web browser) on the 3RS-ENET-CAN. These forwarded data messages are sent when the data become available from serial ports or from CAN interface

The TCP port can be written to, the received frames will be forwarded to output interfaces specified on the routing configuration page #5.

The *Message Header* contains:

4-byte *Axiomatic Tag,* AXIO in capital letters

2-byte *Protocol ID*, 20008 = 0x4E28

2-byte *Message ID*

1-byte *Message Version*, 0 (for future use)

2-byte *Message Data Length*

The proprietary messaging protocol *Message Header* format is presented below.

<table>
<tr><td>Octet
Offset Octet</td><td>0</td><td>1</td><td>2</td><td>3</td></tr>
<tr><td rowspan="3">0</td><td>A</td><td>X</td><td>I</td><td>O</td></tr>
<tr><td>0x41</td><td>0x58</td><td>0x49</td><td>0x4F</td></tr>
<tr><td colspan="4">Axiomatic Tag</td></tr>
<tr><td rowspan="2">4</td><td>0x28</td><td>0x4E</td><td colspan="2" rowspan="2">Message ID</td></tr>
<tr><td colspan="2">Protocol ID (20008)</td></tr>
<tr><td rowspan="2">8</td><td>0x00</td><td colspan="2" rowspan="2">Message Data Length</td><td rowspan="2">Message Data</td></tr>
<tr><td>Message Version=0</td></tr>
</table>

Table 2 – TCP message header format

The *Axiomatic Tag* is used for the message header identification.

The *Protocol ID* defines a proprietary protocol carried by this message. This field allows different protocols to use the same protocol independent message structure. The AX142100 uses Protocol ID = 0x4E28

The *Message ID* defines the type of the Message Data:

Message ID	Message name
0	Undefined message
1	Forwarded data

The first byte of the payload data in the Ethernet frame contains status bits that control how the AX142100 handles the received Ethernet data. In case the Raw data flag is set, all following bytes are considered as data with no special formatting.

<first byte> & 0x40 == Raw data

In case the Raw data flag is not set, the data bytes are considered as CAN data

<first byte> & 0x10 == 0x10 -> 29bit CAN frame ID
<first byte> & 0x10 == 0x00 -> 11bit CAN frame ID
<first byte> & 0x0F == CAN data length

An example printout of the TCP client (see also Table 3 – Example TCP client implementation) reveals the TCP frame contents when a CAN frame is forwarded to Ethernet.

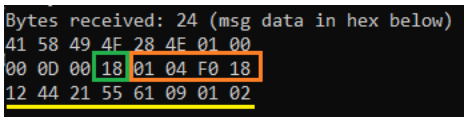


Figure 3 – TCP/IP frame contents (CAN data)

The frame starts with the header bytes described in Table 2. After the header, the first byte of the payload data is marked with green. It contains a flag that the CAN frame has 29bit ID and 8 data bytes. The CAN frame ID is marked with orange and CAN data bytes with yellow.

Please note that the AX142100 considers the Ethernet frames like the one above as CAN data (no Raw data flag set). When this data is forwarded to the CAN interface, the frame ID type (29bit/11bit) and the ID bytes are automatically picked up from the Ethernet frame.

On the other hand, when the Ethernet frame contains Raw data, the data is forwarded ‘as is’ and no special processing is applied (other than the routing rules defined for Ethernet data).

```
#include <winsock2.h>
#include <Ws2tcpip.h>
#include <stdio.h>

#define DEFAULT_BUFLEN        256

#define IP_ADDRESS            "192.168.1.20"
#define FWD_DATA_PORT         4000

#define dRAW_DATA_FLAG        0x40


int main(void) {

    int iResult, index;
    WSADATA wsaData;

    SOCKET ConnectSocket = INVALID_SOCKET;
    struct sockaddr_in clientService;

    int recvbuflen = DEFAULT_BUFLEN;
    char recvbuf[DEFAULT_BUFLEN];

    // Initialize Winsock
    iResult = WSAStartup(MAKEWORD(2,2), &wsaData);
    if (iResult != NO_ERROR) {
        printf("WSAStartup failed with error: %d\n", iResult);
        return 1;
    }

    // Create a socket for connecting to the AX142100
    ConnectSocket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
    if (ConnectSocket == INVALID_SOCKET) {
        printf("socket failed with error: %d\n", WSAGetLastError());
        WSACleanup();
        return 1;
    }

    clientService.sin_family = AF_INET;
    clientService.sin_addr.s_addr = inet_addr( IP_ADDRESS );
    clientService.sin_port = htons( FWD_DATA_PORT );

    // Connect to the AX142100
    iResult = connect( ConnectSocket, (SOCKADDR*) &clientService, sizeof(clientService) );
    if (iResult == SOCKET_ERROR) {
        printf("connect failed with error: %d\n", WSAGetLastError() );
        closesocket(ConnectSocket);
        WSACleanup();
        return 1;
    }

    // Receive data until the AX142100 closes the connection
    do {
        memset((void *)recvbuf, 0x00, sizeof(recvbuf));
        iResult = recv(ConnectSocket, recvbuf, recvbuflen, 0);
        if ( iResult > 0 )
        {
            printf("Bytes received: %d (msg data in hex below)\n", iResult);
            for ( index = 1; index < iResult+1; index++ )
            {
                printf("%02X ", (unsigned char)recvbuf[index-1]);
                if ( (index % 8) == 0 ) printf("\n");
            }
            printf("\n");

            // Send back with ID + 1 (Frame ID is in indexes 12 ... 15)
            if( recvbuf[12] < 255 ) recvbuf[12]++;

            #if 0
            // Send back as raw data instead of CAN data
            recvbuf[11] |= dRAW_DATA_FLAG;
            printf("message flagged as raw data.\n");
            #endif

            iResult = send( ConnectSocket, recvbuf, iResult, 0 );
            if (iResult == SOCKET_ERROR) {
                printf("send failed with error: %d\n", WSAGetLastError());
                closesocket(ConnectSocket);
                WSACleanup();
                return 1;
            }

            printf("%d bytes sent back\n", iResult);
        }
        else if ( iResult == 0 )
            printf("Connection closed\n");
        else
            printf("recv failed with error: %d\n", WSAGetLastError());

    } while( iResult > 0 );

    return 0;
}
```

Table 3 – Example TCP client implementation

The example can be compiled using MinGW: <MinGW location>\bin\gcc.exe -Wall -o data_client data_client.c -lws2_32

5. ECU SETPOINTS ACCESSED WITH AXIOMATIC ELECTRONIC ASSISTANT

This section describes in detail each setpoint, and their default and ranges. The setpoints are divided into setpoint groups as they are shown in EA. For more information on how each setpoint is used by 3RS-ENET-CAN, refer to the relevant section in this user manual.

5.1. J1939 Setpoints

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

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

Table 4 – J1939 Setpoints

If non-default values for the **"ECU Instance Number"** or **"ECU Address"** are used, they will be mirrored during a setpoint file flashing, and will only take effect once the entire file has been downloaded to the unit. After the setpoint flashing is complete, the unit will claim the new address and/or re-claim the address with the new NAME. If these setpoints are changing, it is recommended to close and re-open the CAN connection on EA after the file is loaded so that only the new NAME and address are showing in the J1939 CAN Network ECU list.

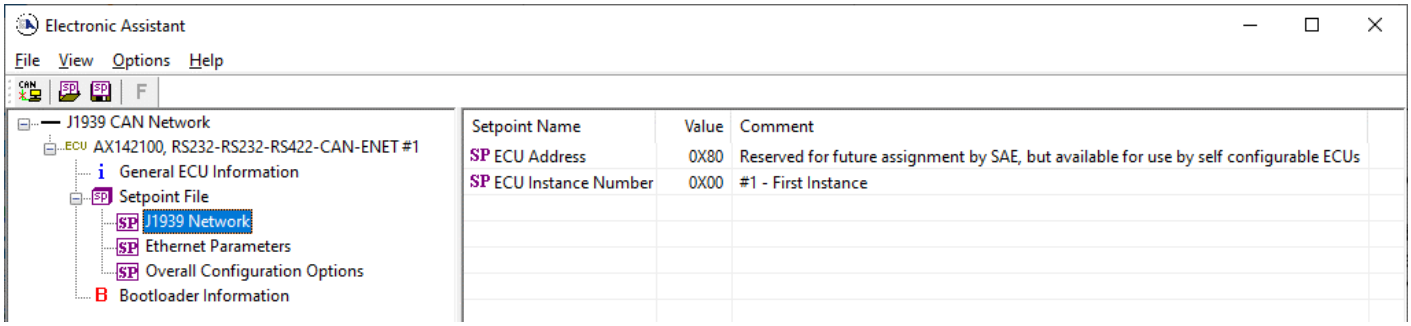


Figure 4 – Screen Capture of J1939 Setpoints

5.2. Ethernet Parameter Setpoints

The Ethernet parameters can be configured using EA. A power cycle is needed for taking the new settings in use.

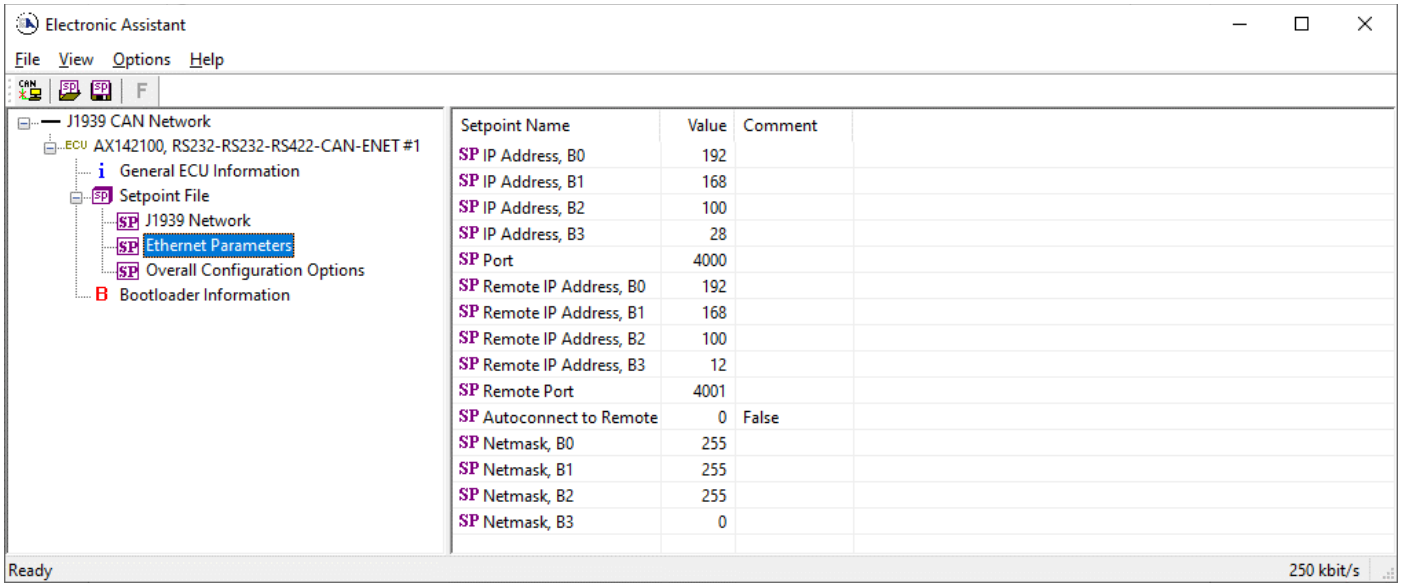


Figure 5 – Screen Capture of Ethernet Parameter Setpoints

Name	Range	Default	Notes
IP Address, B0	0…255	192	These settings define an IP address: 192.168.100.28
IP Address, B1	0…255	168	
IP Address, B2	0…255	100	
IP Address, B3	0…255	28	
Port	0…65535	4000	Default port for incoming TCP connections
Remote IP Address, B0	0…255	192	These settings define an IP address for remote connection: 192.168.100.12
Remote IP Address, B1	0…255	168	
Remote IP Address, B2	0…255	100	
Remote IP Address, B3	0…255	12	
Remote Port	0…65535	4001	Default port for remote TCP connection
Autoconnect to Remote	0, 1	0 – False	Whether to automatically initiate remote TCP connection
Netmask, B0	0…255	255	These settings define a netmask 255.255.255.0
Netmask, B1	0…255	255	
Netmask, B2	0…255	255	
Netmask, B3	0…255	0	

Table 5 – Ethernet Parameter Setpoints

5.3. Overall Configuration Options

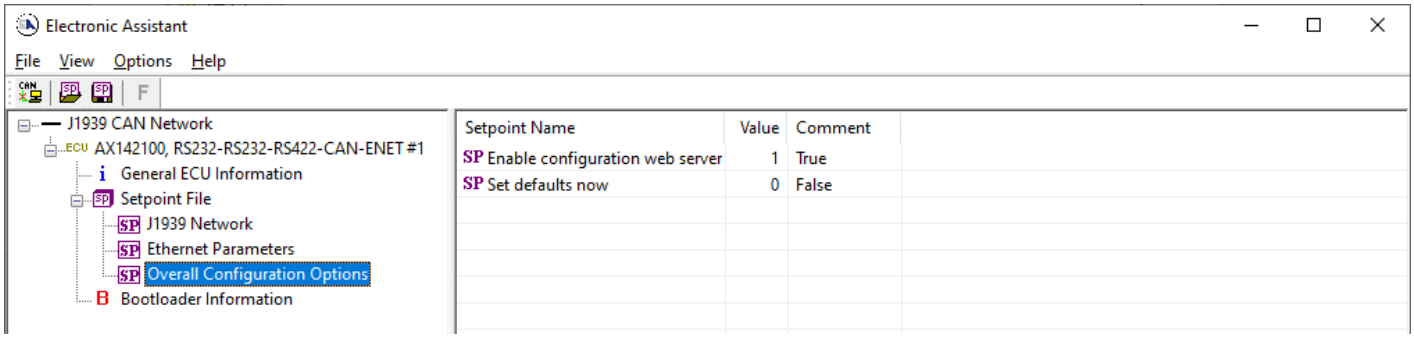


Figure 6 – Screen Capture of Overall Configuration Options Setpoints

Name	Range	Default	Notes
Enable configuration web server	0, 1	1 – True	Configuration web server running on port 80 (TCP)
Set defaults now	0, 1	0 – False	This setpoint is password protected. The password is '**SetDefaults**'.

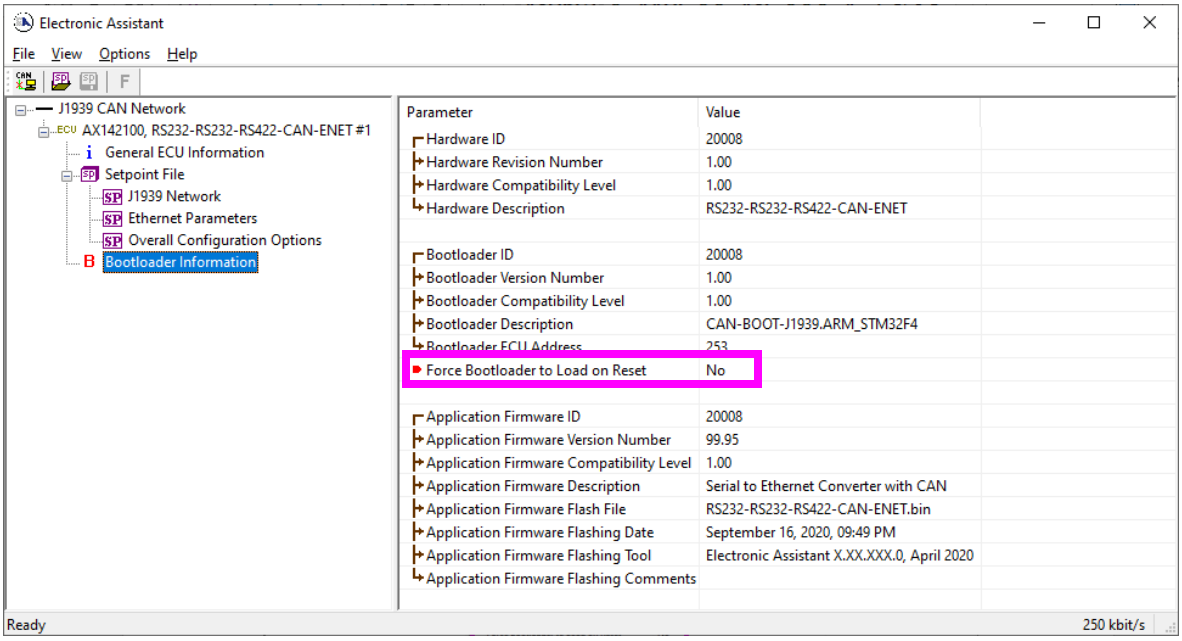
Table 6 – Overall Configuration Options Setpoints

6. REFLASHING OVER CAN WITH EA BOOTLOADER

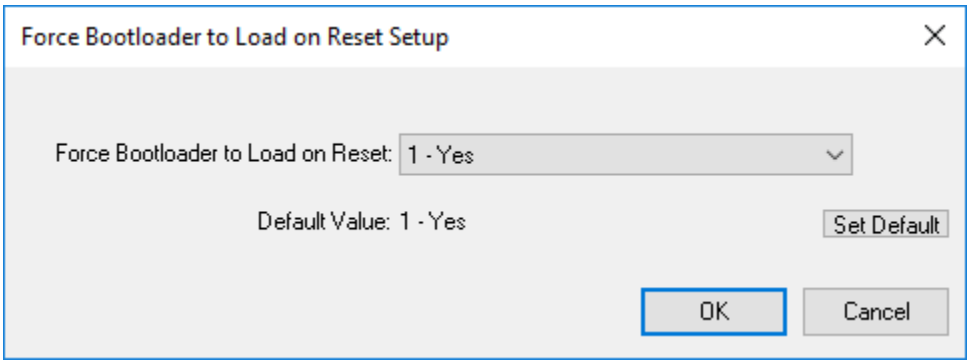
The AX142100 can be upgraded with new application firmware using the **Bootloader Information** section. This section details the simple step-by-step instructions to upload new firmware provided by Axiomatic onto the unit via CAN, without requiring it to be disconnected from the J1939 network.

Note: To upgrade the firmware use Axiomatic Electronic Assistant.

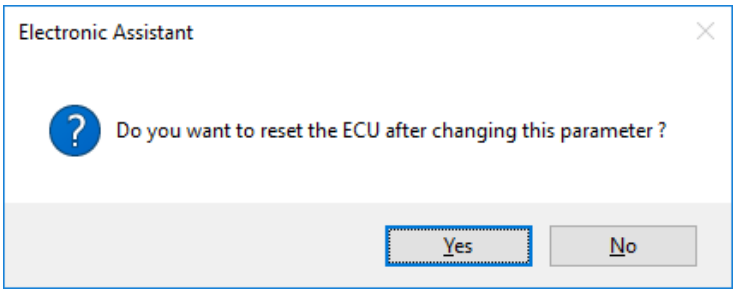
1. When EA first connects to the ECU, the **Bootloader Information** section will display the following information.



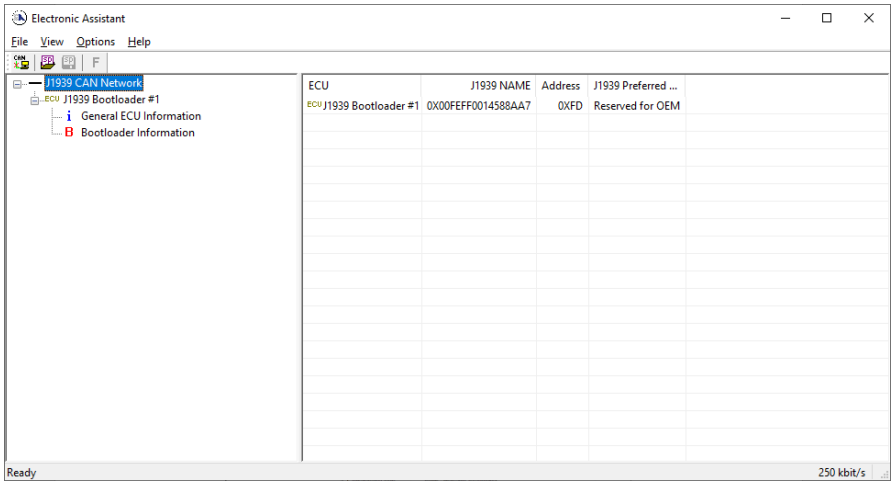
2. To use the bootloader to upgrade the firmware running on the ECU, change the variable **"Force Bootloader To Load on Reset**" to *Yes.*

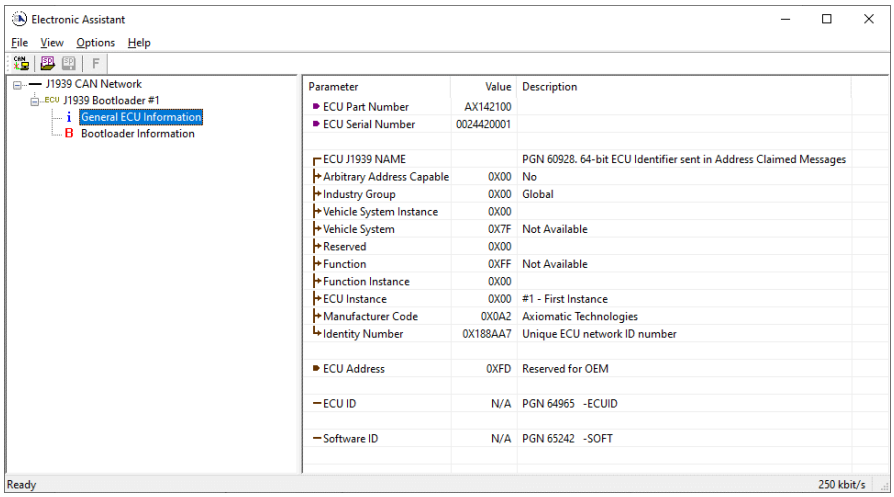


3. When the prompt box asks if you want to reset the ECU, select *Yes*.



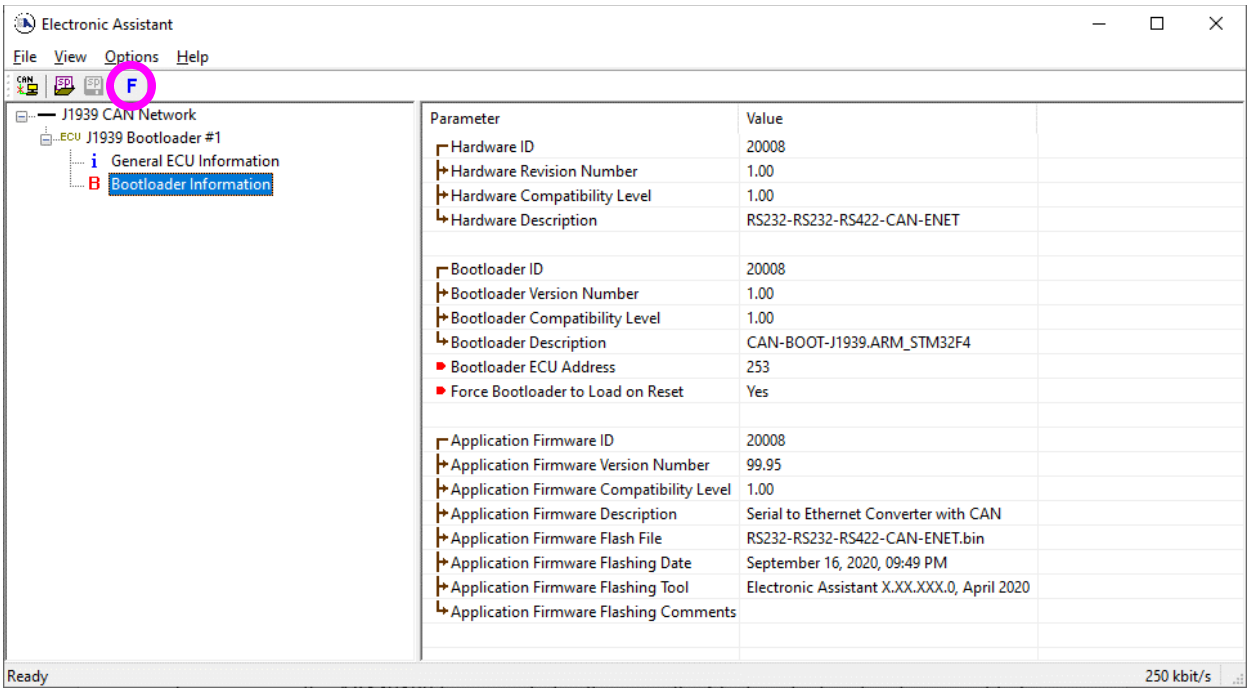
4. Upon reset, the ECU will no longer show up on the J1939 network as an AX142100 but rather as **J1939 Bootloader #1**.





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

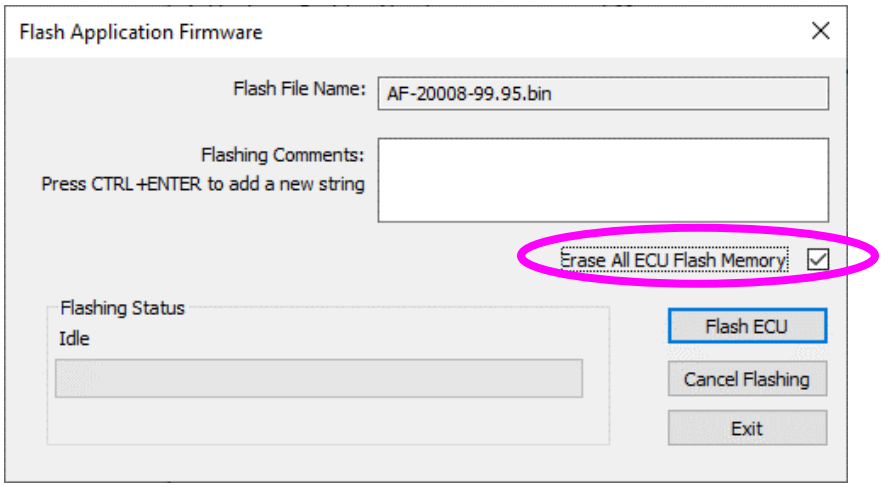
5. When the **Bootloader Information** section is selected, the same information is shown as when it was running the AX142100 firmware, but in this case the **F**lashing feature has been enabled.



6. Select the **F**lashing button and navigate to where you had saved the **AF-20008-x.xx.bin** file sent from Axiomatic. (Note: only binary (.bin) files can be flashed using the EA tool.)

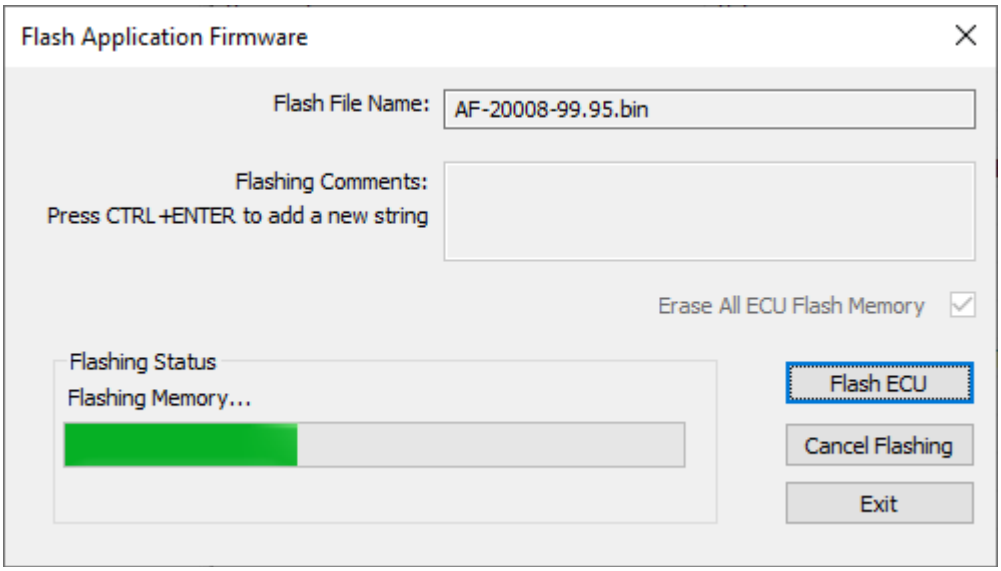
7. Once the Flash Application Firmware window opens, you can enter comments such as “Firmware upgraded by [Name]” if you so desire. This is not required, and you can leave the field blank if you do not want to use it.

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

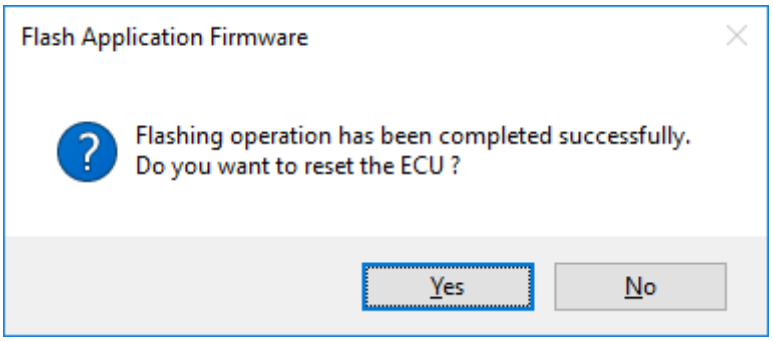


NOTE: It is good practice to tick the “Erase All ECU Flash Memory” box. Please note, that selecting this option will **erase ALL data stored in non-volatile flash**. It will also erase any configuration of the setpoints that might have been done to the ECU and reset all setpoints to their factory defaults. In case the controller contains custom settings, those settings need to be saved to PC before reflashing.

A progress bar will show how much of the firmware has been sent as the upload progresses. The more traffic there is on the J1939 network, the longer the upload process will take.



Once the firmware has finished uploading, a message will pop up indicating the successful operation. If you select to reset the ECU, the new version of the AX142100 application will start running, and the ECU will be identified as such by EA. Otherwise, the next time the ECU is power-cycled, the AX142100 application will run rather than the bootloader function.



Note: If at any time during the upload the process is interrupted, the data is corrupted (bad checksum) or for any other reason the new firmware is not correct, i.e. bootloader detects that the file loaded was not designed to run on the hardware platform, the bad or corrupted application will not run. Rather, when the ECU is reset or power-cycled the **J1939 Bootloader** will continue to be the default application until valid firmware has been successfully uploaded into the unit.

APPENDIX A - TECHNICAL SPECIFICATION

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/.

Specifications are typical at nominal input voltage and 25 degrees C unless otherwise specified.

Power

Power Supply Input - Nominal	12 V or 24 Vdc nominal; 9…36 Vdc
Under-voltage Protection	Hardware shuts down at 6Vdc.
Surge Protection	95 Vdc
Over-voltage Protection	Hardware shuts down at 45Vdc.
Reverse Polarity Protection	Provided up to -90V

Functionality

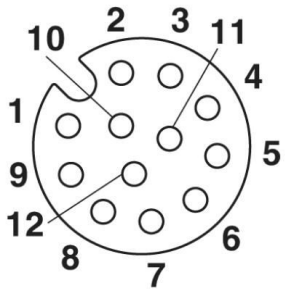
Conversion Platform	The Protocol Converter comes pre-programmed with standard protocol conversion logic for bidirectional data exchange between Ethernet (proprietary TCP communications), an RS-422 bus, two RS-232 buses and a CAN network (SAE J1939). Data is forwarded "as-is" between the different serial ports. Also, CAN / Ethernet data is forwarded directly to serial interfaces with the configuration allowing the user to specify the CAN message ID (or TCP port) to listen for data to be forwarded.
Ethernet	Single 10/100 Mbit Ethernet compliant port 10BASE-T, 100BASE-Tx (auto-negotiation and full-duplex supported) Auto-MDIX
RS-422	Single RS-422 port Baud rate: up to 10.5 MBit/s Note: RS-422 connections can be used as RS-485. See pinout.
RS-232	Two RS-232 ports for serial communications Three-wire Baud rate: up to 400 kBit/s
ASCII Features	Maximum Number of ASCII devices: 2 Serial Communications Port 0: RS422 Serial Communications Port 1: RS232 Message Queue Size: Configurable
CAN	Single SAE J1939 port Isolated Baud rate: 250 kBit/s (default) (CANopen® model available)

General Specifications

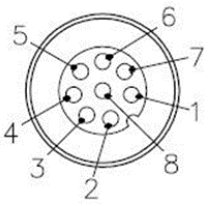
Microcontroller	STM32F407VGT7 32-bit, 1Mbyte Flash Program Memory
Quiescent Current Draw	70 mA @12 V; 40 mA @24 V Typical
Isolation	CAN isolation: 330 Vrms
LED Indicators	POWER LED GREEN= Power ON RED = Fault condition GREEN/RED = Power OFF 2 GREEN for Ethernet LEDs LINK/ACT: ON means connection (LINK) Flashing means activity (ACT) OFF means Ethernet connection is down Transmission Speed: 100 Mbit/s = ON 10 Mbit/s = OFF
Web Interface	Available Refer to the User Manual. The functionality of the web interface includes but is not limited to the following. • Specify CAN message filters and CAN message IDs to be received • Link RS-232 or RS-422 to CAN bus and Ethernet • Define CAN node ID, and baud rate • Define Ethernet parameters (IP address, netmask) • Configure message queues

<table>
<tr><td>User Interface</td><td>Axiomatic Electronic Assistant KIT P/Ns: AX070502 or AX070506K
The Axiomatic Electronic Assistant for Windows operating systems comes with a royalty-free license for use on multiple computers. It requires an Axiomatic USB-CAN converter to link the device's CAN port to a Windows-based PC.
The functionality of the Axiomatic Electronic Assistant includes:
IP address configuration
Firmware Reflashing</td></tr>
<tr><td>Operating Conditions</td><td>-40 to 70°C (-40 to 158°F)</td></tr>
<tr><td>Weight</td><td>0.15 lb. (0.068 kg) without mating cables</td></tr>
<tr><td>Enclosure and Dimensions</td><td>Nylon 6/6, 30% glass fill
UL 94V-0
Ultrasonically welded
Refer to dimensional drawing.</td></tr>
<tr><td>Protection Rating</td><td>IP67; Unit is encapsulated within the housing.</td></tr>
<tr><td>Vibration</td><td>Random Vibration: Z-axis tracked vehicle profile (5 hr/axis in all 3 axes)
Sinusoidal Component: 8.9 G Sine sweep, 2.5 hr/axis in all 3 axes</td></tr>
<tr><td>Shock</td><td>50 g, 5 impacts per test, 6-20 ms impact duration</td></tr>
<tr><td>Electrical Connections</td><td>CAN / RS-232 / RS-422 Connector
1 Phoenix Contact M12 12-pin connector (A-coded), Female P/N: 1441833

Note: To use RS-422 as RS-485, connect the Tx+ and Rx+ pin to D+ on the RS-485 connector. Also connect the Tx- and Rx- to pin D-.

<table>
<tr><th>PIN#</th><th>Description</th></tr>
<tr><td>1</td><td>RS-422 RX+</td></tr>
<tr><td>2</td><td>RS-422 TX+</td></tr>
<tr><td>3</td><td>RS-422 RX-</td></tr>
<tr><td>4</td><td>RS-232 TX 2</td></tr>
<tr><td>5</td><td>RS-232 RX 2</td></tr>
<tr><td>6</td><td>CAN_L</td></tr>
<tr><td>7</td><td>CAN_H</td></tr>
<tr><td>8</td><td>RS-232 TX 1</td></tr>
<tr><td>9</td><td>RS-232 RX 1</td></tr>
<tr><td>10</td><td>RS-422 TX-</td></tr>
<tr><td>11</td><td>GND</td></tr>
<tr><td>12</td><td>GND</td></tr>
</table>


Ethernet / Power Connector
1 Phoenix Contact M12 8-pin connector (A-coded), Female, P/N: 1441817

<table>
<tr><th>PIN#</th><th>Description</th></tr>
<tr><td>1</td><td>Power +</td></tr>
<tr><td>2</td><td>Power -</td></tr>
<tr><td>3</td><td>Power -</td></tr>
<tr><td>4</td><td>Ethernet TX-</td></tr>
<tr><td>5</td><td>Ethernet RX+</td></tr>
<tr><td>6</td><td>Ethernet TX+</td></tr>
<tr><td>7</td><td>Power +</td></tr>
<tr><td>8</td><td>Ethernet RX-</td></tr>
</table>

</td></tr>
<tr><td>Mating Connectors</td><td>Not supplied
Mating connectors should meet the following standard for M12 Connectors, IEC 61076-2-101:2012. They should be A-coded.</td></tr>
<tr><td>Mating Cables</td><td>Refer to the drawings for AX070533 and AX070531. They are located on www.axiomatic.com under the log-in tab in the user manual section.
AX070531: Ethernet and Power Cable 1.7 m (5.5 ft.), 8-pin M12 A-coded, Unterminated Leads, Ethernet Jack
AX070533: CAN, RS-232, RS-422 Cable 1.5 m (5 ft.), 12-pin M12, Unterminated Leads</td></tr>
<tr><td>Installation</td><td>Suits two M5 or #10 mounting bolts.</td></tr>
</table>

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

OUR PRODUCTS

AC/DC Power Supplies

Actuator Controls/Interfaces

Automotive Ethernet Interfaces

Battery Chargers

CAN Controls, Routers, Repeaters

CAN/WiFi, CAN/Bluetooth, Routers

Current/Voltage/PWM Converters

DC/DC Power Converters

Engine Temperature Scanners

Ethernet/CAN Converters, Gateways, Switches

Fan Drive Controllers

Gateways, CAN/Modbus, RS-232

Gyroscopes, Inclinometers

Hydraulic Valve Controllers

Inclinometers, Triaxial

I/O Controls

LVDT Signal Converters

Machine Controls

Modbus, RS-422, RS-485 Controls

Motor Controls, Inverters

Power Supplies, DC/DC, AC/DC

PWM Signal Converters/Isolators

Resolver Signal Conditioners

Service Tools

Signal Conditioners, Converters

Strain Gauge CAN Controls

Surge Suppressors

OUR COMPANY

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

QUALITY DESIGN AND MANUFACTURING

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

WARRANTY, APPLICATION APPROVALS/LIMITATIONS

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

COMPLIANCE

Product compliance details can be found in the product literature and/or on axiomatic.com. 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