User Manual UMAX140900RST
Version 1.1
Firmware 1.xx

USER MANUAL

Ethernet to CAN Converter

P/N: AX140900RST

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

A	Ampere
ARP	Address Resolution Protocol
°C	Celsius (degree)
CAN	Controller Area Network
CE	Conformité Européenne (European Conformity)
EA	The Axiomatic Electronic Assistant PC application software
EEPROM	Electrically Erasable Programmable Read-Only Memory
EMC	Electromagnetic Compatibility
ESD	Electrostatic Discharge
FCC	Federal Communications Commission
G	Acceleration in Gravity Units
GPL	General Public License
hr	hour
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
ID	Identifier
IEC	International Electrotechnical Commission
IP	Internet Protocol or Ingress Protection (for housing)
ISO	International Organization for Standardization
L	Length (for size)
LAN	Local Area Network
LED	Light-Emitting Diode
m	meters
MAC	Media Access Control (address)
MDIX	Medium Dependent Interface Crossover (MDI-X)
ms	millisecond
PA	Polyamide
PHY	Physical Layer Transceiver (Ethernet chip)
P/N	Part Number
PoE	Power Over Ethernet
RoHS	Restriction of Hazardous Substances
RTOS	Real-Time Operating System
SP	Service Pack
SSP	Software Support Package
SW	Software
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UL	Underwriters Laboratories (safety organization)
USB	Universal Serial Bus
V	Volt
VDC	Volt Direct Current
W	Watt or Width (for size)
WAN	Wide Area Network

TABLE OF CONTENTS

1 INTRODUCTION

The following user manual describes architecture, functionality, and configuration parameters of the Ethernet to CAN Converter. It also contains technical specifications and installation instructions of the converter.

The user should check whether the application firmware installed in the converter is covered by this user manual. It can be done using any web browser connected to the converted over the Ethernet connection, see: Configuration Parameters section for more details.

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 application firmware version 1.xx. Updates specific to the user manual are done by adding letters: A, B, …, Z to the user manual version number.

2 CONVERTER DESCRIPTION

The Ethernet to CAN Converter is a simple device converting CAN frames into UDP or TCP IP datagrams and sending them over the Ethernet network. The device can also convert UDP or TCP datagrams into CAN frames.

The converter has one CAN and one Ethernet port. It supports a high-speed CAN with baud rate up to 1Mbit/s and a fast 100Mbit/s Ethernet. All standard and extended CAN frames, including data and remote frames, are supported.

The power can be passed through to the CAN port. Protection is provided.

The converter contains a web server to setup configuration parameters and monitor the internal state of the converter using a web browser. The user can also update the converter firmware using the web browser.

A simple command-line `AxioDisc.exe` Windows application is provided to locate a converter on the LAN.

To ensure low latency in processing CAN and Ethernet messages, the converter software runs under control of a real-time operating system.

The converter is designed to work on off-road machinery or in a harsh industrial environment with power transients, high humidity, and vibrations.

2.1 Hardware Block Diagram

The converter hardware block diagram is presented in Figure 1.

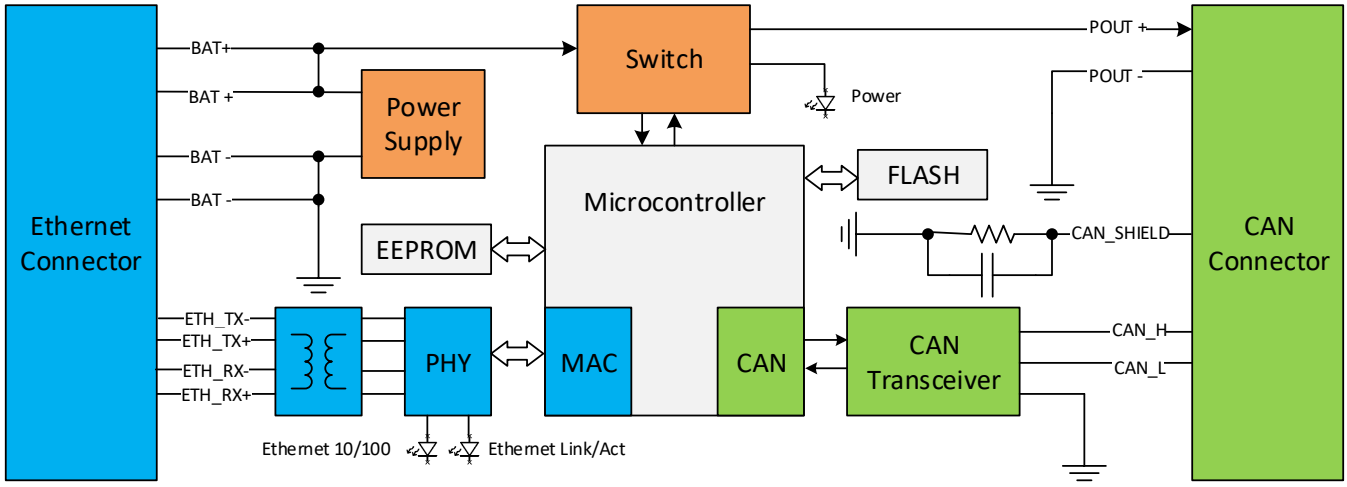


Figure 1. The Converter Hardware Block Diagram

The converter is powered from the Ethernet connector using dedicated power lines. The power from the Ethernet connector can be delivered to the CAN connector through a switch controlled by a microcontroller.

The Ethernet transformers with PHY, together with the CAN transceiver, provide Ethernet and CAN physical interfaces.

A powerful ARM Cortex-M4 microcontroller runs IP protocol stack and all Ethernet to CAN conversion logic.

2.2 LED Indicators

There are three LED indicators on the front panel of the converter. A bi-color “Power” indicator shows whether the unit is powered. It turns red when there is an error on the CAN power output.

The two Ethernet LED indicators are hardwired to the PHY chip and show the transmission speed “Ethernet 10/100” and the link/activity status “Ethernet Link/Act”, see Table 1:

Table 1. Converter LED Indicators

Name	Color	Description	
Power	Green/Red	Off Green Red	The converter is not powered. The converter is powered. CAN Power Output is in normal condition. CAN Power Output is in a fault condition.
Ethernet 10/100	Green	On Off	Ethernet speed is 100 Mbit/s Ethernet speed is 10 Mbit/s
Ethernet Link/Act	Green	On Off Blinking	Ethernet link is up Ethernet link is down Ethernet link is up and active

2.3 Firmware Organization

The Ethernet to CAN Converter firmware contains two independent parts: the *Communication Device* and the *Web Server*, see Figure 2:

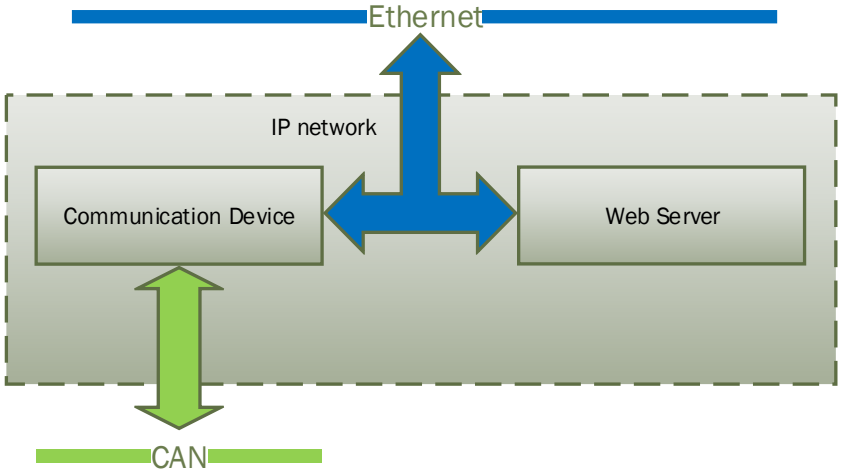


Figure 2. Converter Firmware Architecture

The *Communication Device* is responsible for the protocol conversion between CAN and Ethernet networks and the *Web Server* provides the converter user interface.

Both the *Communication Device* and the *Web Server* use the same IP network interface. The IP address resolution is provided by the ARP protocol.

2.3.1 Communication Device

The *Communication Device* utilizes a proprietary communication protocol to communicate CAN messages and other auxiliary information over the Ethernet. It supports a client/server communication model. In this model, the *Communication Device* has a primary server role, allowing external clients to establish independent connections with the device.

In addition to the server role, the device can also act as a client, if the *Auto Connect to Remote* configuration parameter is set to *Yes*. In this case, the device will try to establish a connection with a customer specified remote server.

The total number of remote connections is limited to 10. If the CAN network traffic is high, this number should be further reduced, or the connections will become unstable due to limited internal resources of the microcontroller, which are dynamically allocated between open connections.

The *Communication Device* can use either UDP or TCP internet protocol (IP), depending on the value of the *Device Port Type* configuration parameter.

2.3.1.1 UDP Protocol

The UDP protocol is set by default. Since it is a connectionless protocol, one data socket serves all device communication needs, see Figure 3.

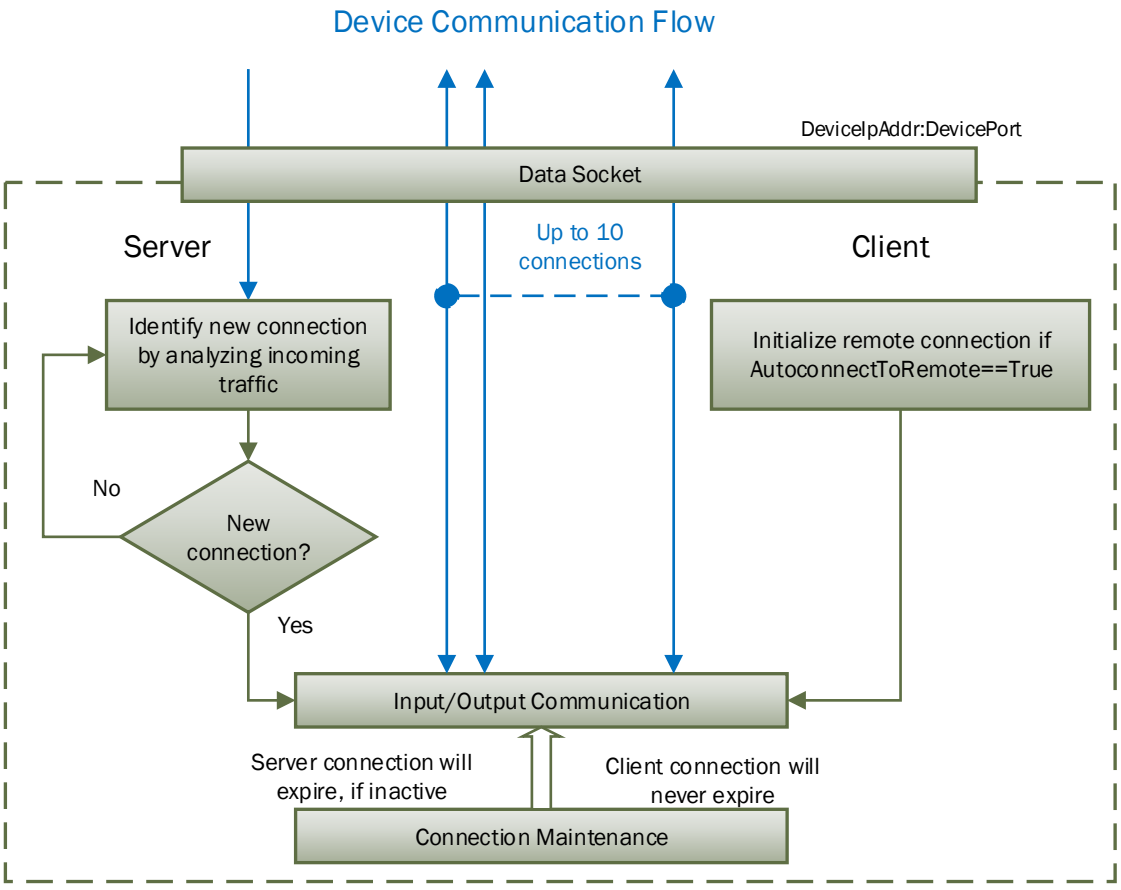


Figure 3. Communication Device. UDP Protocol

All connections with the device are virtual. On the server side, the device analyzes the incoming traffic to check for the new connections. Once a new IPAddress:Port combination is detected, the connection is established and the device starts sending CAN data with Heartbeat messages to the new node.

There are no restrictions on the IP address and port for the incoming connections.

If a client-side is activated by the *Autoconnect to Remote* configuration parameter, the device will automatically start sending CAN data with Heartbeat messages to the remote node on start-up.

To ensure that the device does not send data to not functioning (dead) or disconnected nodes, the server side connections will expire in 10 sec of inactivity, when no valid data is received from the remote node. The client-side connection will never expire.

2.3.1.2 TCP Protocol

When TCP protocol is used, the *Communication Device* opens an individual data socket for each device connection, see: Figure 4.

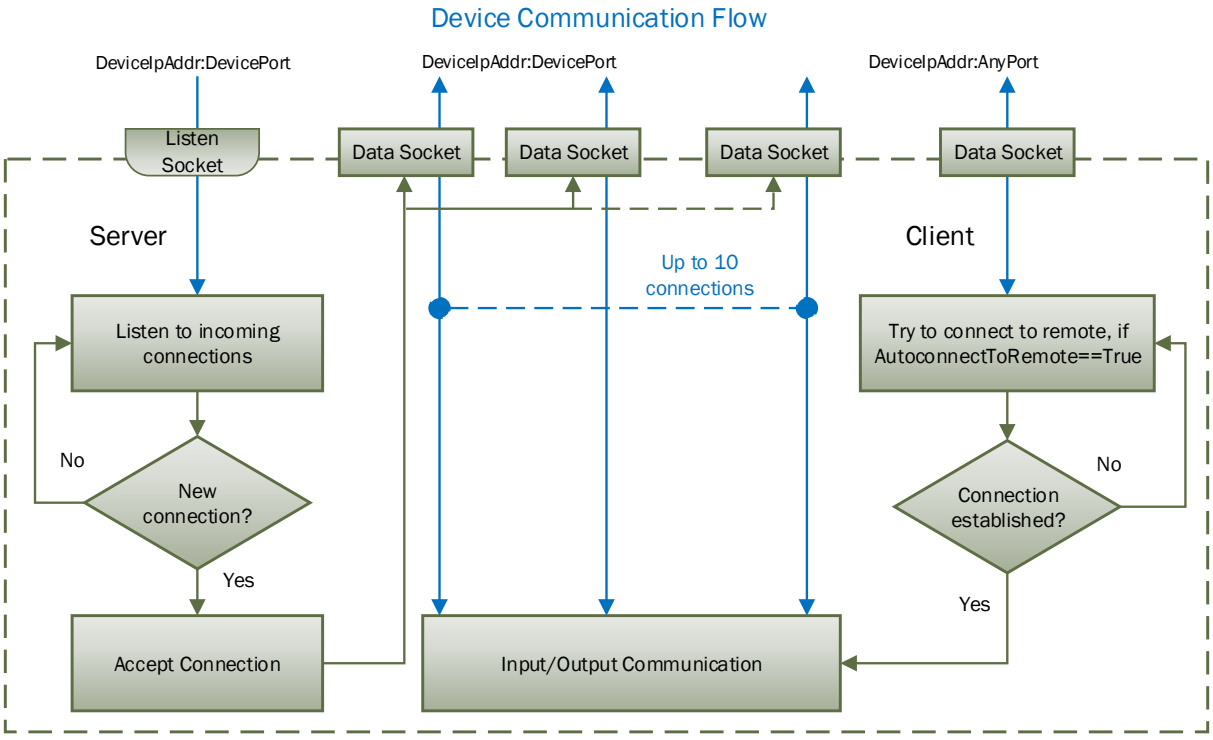


Figure 4. Communication Device. TCP Protocol

The server side opens a listening socket for incoming connections. Once a connection is accepted, a new data socket is created to handle input/output communication with the remote node. There are no restrictions on the IP address and port for the incoming connections, similar to the UDP mode.

On the client side, if *Auto Connect to Remote* is set to *Yes*, a data socket is created for connection with the remote node. A random free port number is assigned to the socket. If the connection drops, the device will try to automatically reconnect with the node to maintain the client connection.

2.3.2 Web Server

The *Web Server* provides a user front-end interface with the converter. It runs a dynamic website that shows: the converter general information, configuration parameters, and the converter real-time diagnostics.

The user can also change configuration parameters and upload the new firmware through this website.

The web browser should support JavaScript.

2.3.3 Firmware Updates

The firmware can be remotely updated through the web server and supports RESTful API.

2.3.4 Network Discovery

The firmware supports a proprietary Axiomatic discovery protocol. It allows to locate a converter with unknown IP address and/or web server port on a LAN using a simple Axiomatic discovery application `AxioDisc.exe`.

3 CONVERTER CONFIGURATION

The converter supports configuration over the internal website running on the device embedded web server. The default *Device IP Address* is *192.168.0.34* and the default *Web Server Port* is *80*. Please, make sure that there are no other devices on this IP address when connecting the converter for the first time to your LAN for configuration.

To connect to the device, run any web browser and point it to the *Device IP Address*. It is not necessary to specify the *Web Server Port* if the web server uses a standard port *80*.

After a successful connection, you will see the device home page, see: Figure 5.

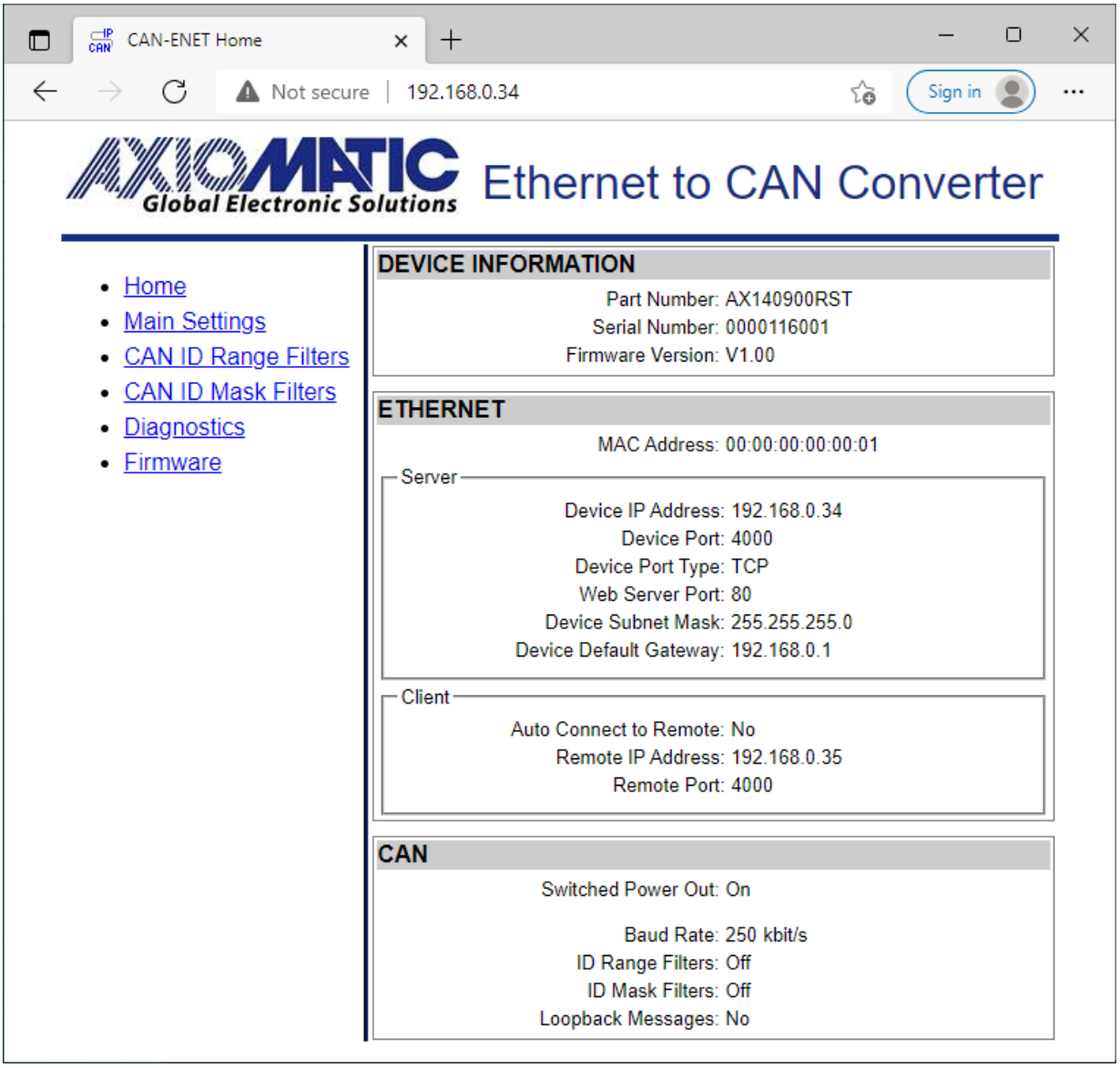


Figure 5. Converter Home Page

The home page shows the device information, including the converter part number, serial number, and firmware version. It also shows *Ethernet* and *CAN* configuration parameters including the status of CAN ID range and mask input filters.

You will need to allow the site to run JavaScript (this setting is default in most web browsers). If JavaScript is disabled, the website will show a message asking to activate JavaScript at the top of the web page, see Figure 6.

Figure 6. Enable JavaScript Prompt

The *Ethernet* configuration parameters are combined into *Server* and *Client* groups for convenience.

The *Ethernet* and *CAN* configuration parameters have tooltips clarifying their meaning, see Figure 7.

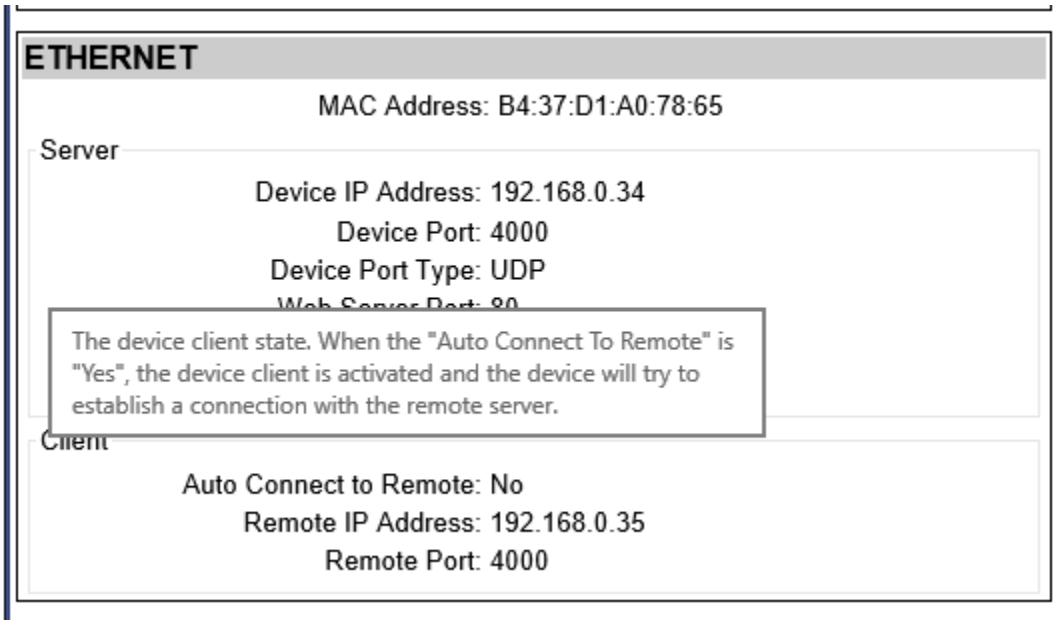


Figure 7. A Tooltip for the "Auto Connect to Remote" Configuration Parameter

3.1 Changing Configuration Parameters

All configuration parameters except the CAN ID range and mask filter settings can be changed through the *Main Settings* web page. The CAN ID range and mask filters have their own configuration pages: *CAN ID Range Filters* and *CAN ID Mask Filters*, see Figure 8.

The configuration pages can be reached by clicking on their links on the left side of the website.

Each configuration web page has fields to enter values of the configuration parameters and three buttons: *Save Settings*, *Discard Settings* and *Set Defaults*.

The *Save Settings* button will save configuration parameters to non-volatile memory and apply the new settings. The appropriate converter subsystems will be restarted without rebooting of the whole converter. The rebooting is available from the *Diagnostics* screen.

The *Discharge Settings* button will bring back the original converter settings before editing, and the *Set Defaults* button will load the default values of the configuration parameters into the data fields on the page.

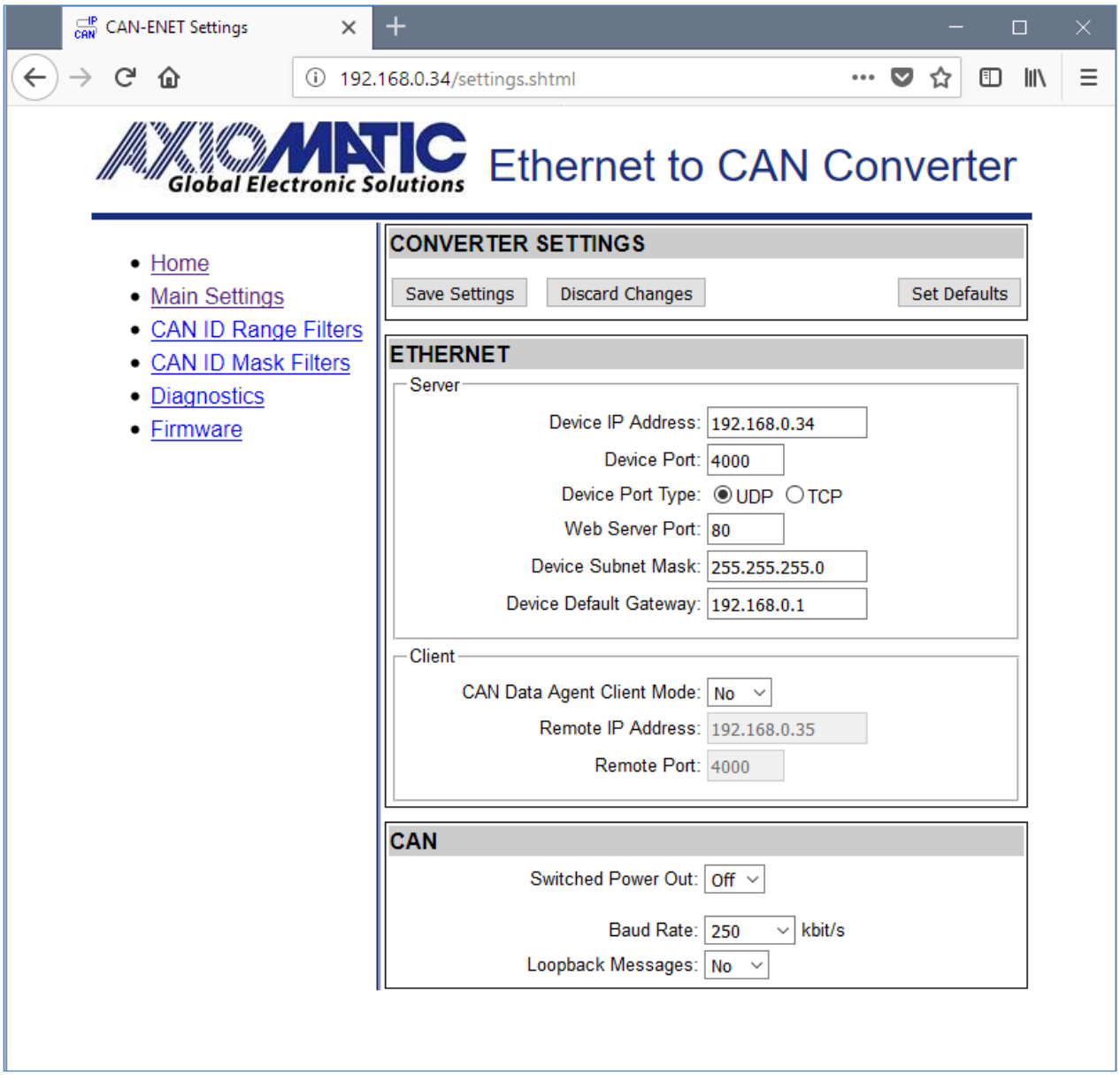


Figure 8. Converter Main Settings Page

The configuration parameters have tooltips for the user convenience. The *Remote IP Address* and *Remote Port* are disabled when *Auto Connect to Remote* is set to *No*.

The page runs a script to check the validity of the new configuration parameters before uploading them to the web server. For example, the following alert message will be displayed if the user enters the same value for the *Device Port* and the *Web Server Port*:

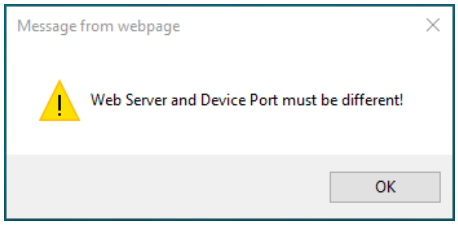


Figure 9. Settings Alert. Error in Configuration Parameters

The website messages should be enabled (not suppressed) in the browser to see this and other feedback messages.

After pressing the *Save Settings* button and saving the configuration parameters, the converter replies with a confirmation message showing a result of the saving operation. For example, if the user has successfully changed the *Device Port* value, the following message will appear:

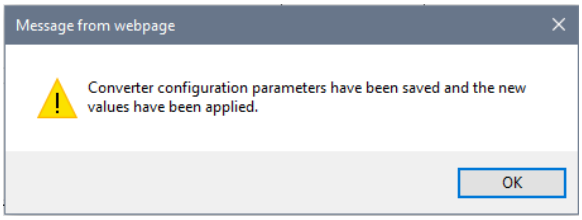


Figure 10. Settings Alert. Configuration Parameters have been Changed Successfully

The new configuration parameters are applied immediately after saving. The converter performs all necessary internal reconfigurations and resets on the fly. If the user changed the *Device IP Address* or the *Web Server Port*, the website will be automatically re-loaded at the new location.

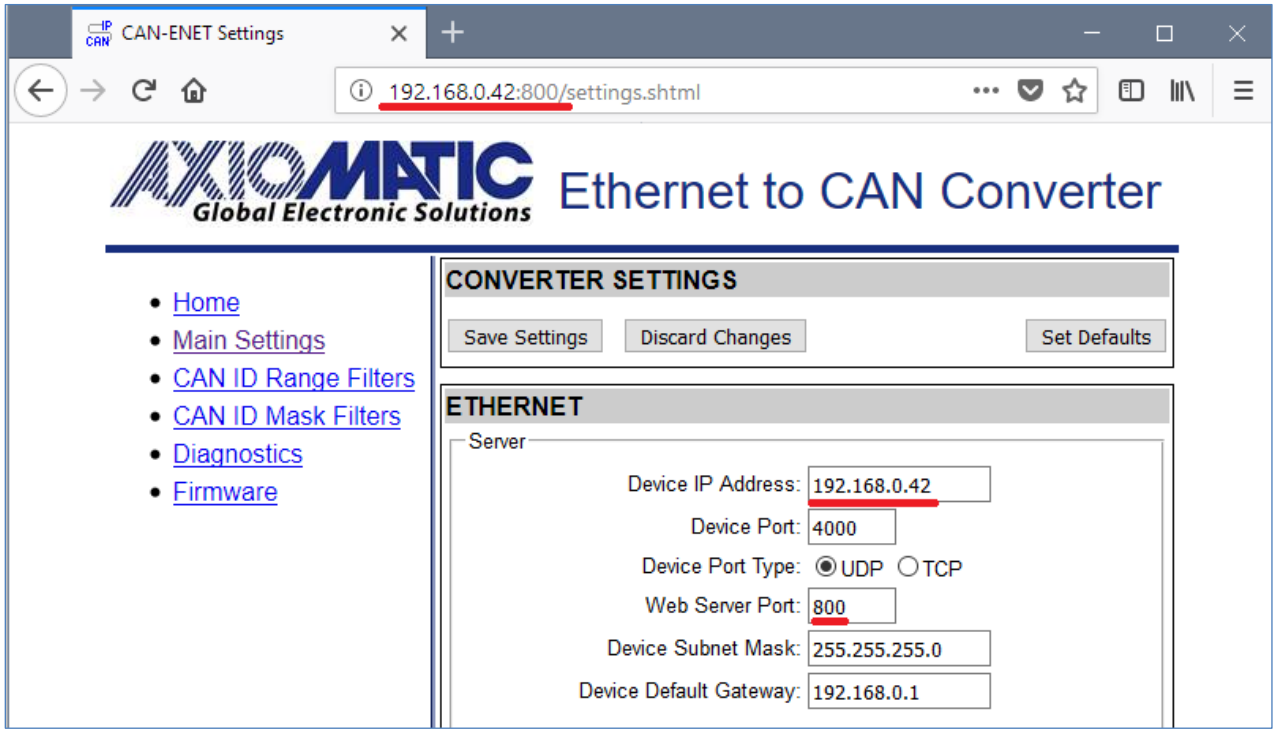


Figure 11. Website Automatic Relocation

In case the user leaves the page without saving, all changes will be discarded. The user can also discard changes by pressing the *Discard Changes* button.

3.2 Ethernet Configuration

All *Ethernet* configuration parameters can be changed through the *Main Settings* web page, except the *MAC Address,* which is programmed at the factory. The user-changeable configuration parameters are presented in Table 2.

The user should avoid using special IP addresses (broadcast, multicast, loopback, etc.) when configuring the *Device IP Address* since this can lead to a permanent loss of communication with the embedded web server.

Table 2. Ethernet Configuration Parameters

Configuration Parameter	Default Value	Range	Description
Device IP Address	192.168.0.34	Any IP address[1]	The device IP address. The embedded web server uses the same IP address.
Device Port	4000	Any port value except the *Web Server Port* and the *Discovery Protocol Port*	The device server port. The device is listening to this port for incoming connections. The *Discovery Protocol Port* (35100) and the *Web Server Port* should not be used.
Device Port Type	UDP	{UDP, TCP}	Type of the IP protocol used by the device. The device server and client use the same IP protocol.
Web Server Port	80	Any port value except the *Device Port* and the *Discovery Protocol Port*	The communication port of the embedded web server.
Device Subnet Mask	255.255.255.0	Any IP address[1]	The device subnet mask. Used also by the embedded web server.
Device Default Gateway	192.168.0.1	Any IP address[1]	The device default gateway. Used also by the embedded web server.
Auto Connect to Remote	No	{No, Yes}	The device client state. When the *Auto Connect to Remote* is *Yes*, the device client is activated, and the device will try to establish a connection with the remote server.
Remote IP Address	192.168.0.35	Any IP address	The remote server IP address. Used by the device client when the *Auto Connect to Remote* is *Yes*.
Remote Port	4000	Any port value	The remote server port. Used by the device client when the *Auto Connect to Remote* is *Yes*.

[1]Must be assigned by the network administrator.

3.3 CAN Configuration

The CAN configuration parameters can be changed through the *Main Settings*, *CAN ID Range Filters,* and *CAN ID Mask Filters* web pages.

The main CAN configuration parameters are available through the *Main Settings* web page, see Table 3.

Table 3. Main CAN Configuration Parameters

Configuration Parameter	Default Value	Range	Description
Switched Power Out	Off	{Off, On}	State of the switch delivering power to the CAN connector.
Baud Rate	250 kbit/s	{1000, 666.6(6), 500, 250, 125, 100, 83.3(3), 50, 20, 10}[1]	The CAN baud rate.
Loopback Messages	No	{No, Yes}	Specifies, whether the messages received over the Ethernet and transmitted on the CAN bus, are sent back to the Ethernet. Setting this value to *Yes* can create an eternal loop when the same messages are bounced between two or more converters. Use with caution.

[1] *666.6(6) and 83.3(3) kbit/s are set as 667 and 83 kbit/s, respectively, in the drop-down menu. The 666.6(6) kbit/s is absent in firmware V1.xx…4.xx.*

The CAN filters have their own web pages for setting configuration parameters. If all filters are disabled, all input CAN messages will be output on the Ethernet network.

3.3.1 CAN ID Range Filters

The CAN ID range filters are set through the *CAN ID Range Filters* configuration web page.

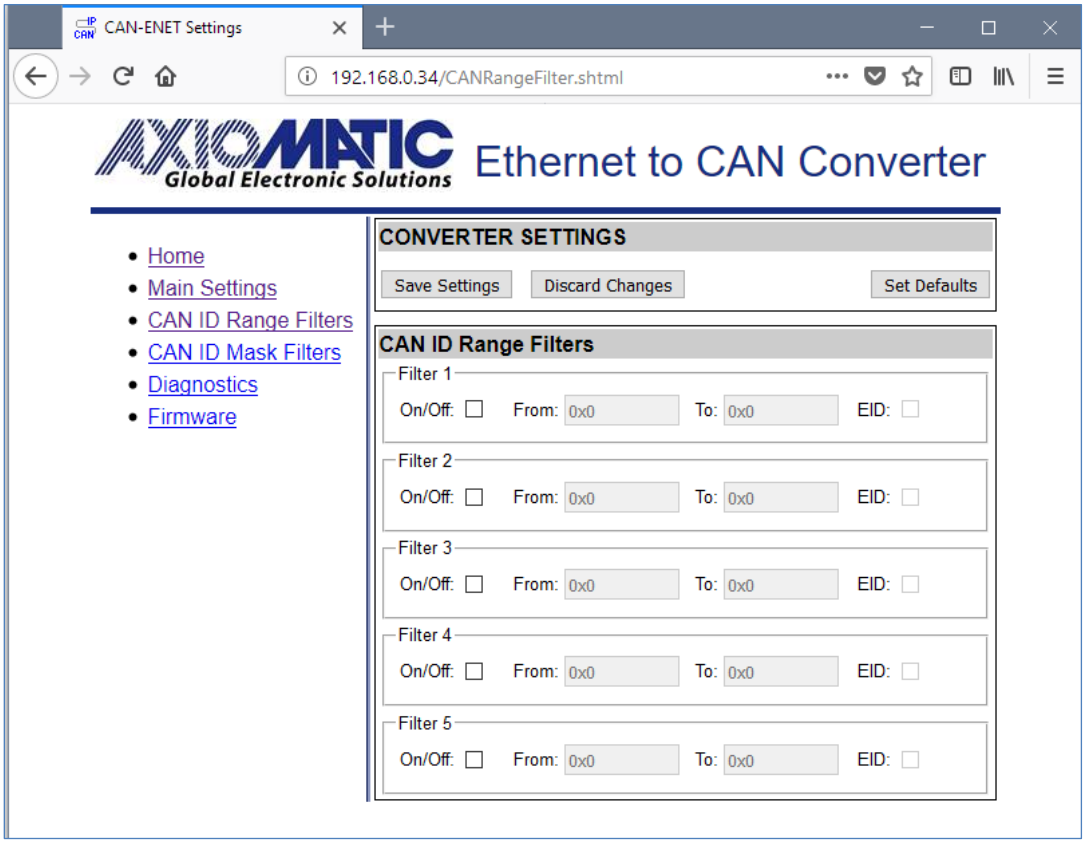


Figure 12. Converter CAN ID Range Filters Page

The user can independently configure five CAN ID range filters: *Filter 1, Filter 2, …, Filter 5.*

Once the filter is activated by checking the *On/Off* box, the CAN input messages will pass through to the Ethernet network only if their CAN ID is within the range specified by *From* and *To* configuration parameters.

$$If\ ID_{CAN} \in [From;\ To],\ then\ the\ message\ is\ accepted. \qquad (1)$$

Where: ID_{CAN} – CAN message ID,
$From$ – *From* configuration parameter,
To – *To* configuration parameter.

The *EID* box (*Extended ID* box) defines whether the CAN message ID is regular or extended.

All CAN ID range filters run in parallel. It is sufficient to satisfy requirements of any active filter to pass the CAN message to the Ethernet network.

If no active filters are defined, it is considered that the CAN ID range filters are disabled, and do not participate in the message filtering process. In this case, *ID Range Filters* are *Off* on the home page, and, if other filters are also disabled, all CAN input messages will be sent to the Ethernet network.

3.3.2 CAN ID Mask Filters

The CAN ID mask filters are set through the *CAN ID Mask Filter* configuration web page.

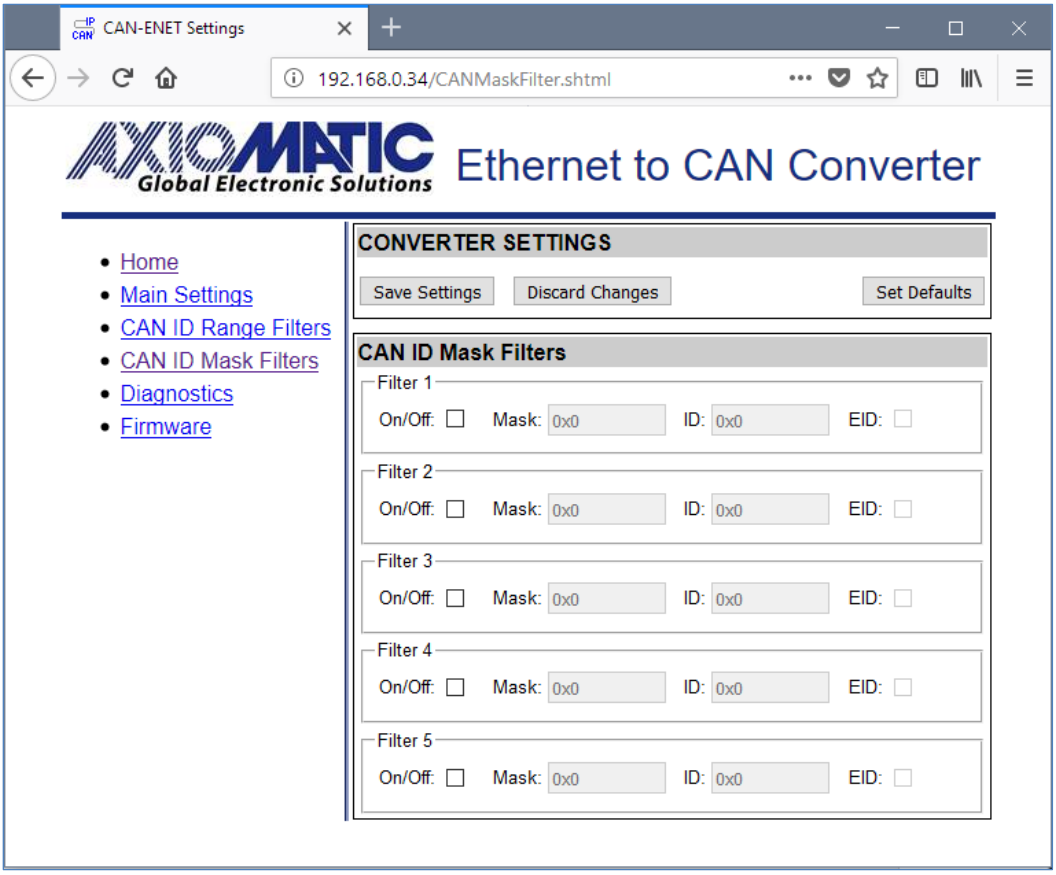


Figure 13. Converter CAN ID Mask Filters Page

There are five independent CAN ID mask filters: *Filter 1, Filter 2, …, Filter 5* available to the user.

Once the filter is activated by checking the *On/Off* box, the CAN input messages will pass through the filter to the Ethernet network only if their CAN ID satisfies the following condition:

$$If\ ID = ID_{CAN}\ \&\ Mask,\ then\ the\ message\ is\ accepted. \qquad (2)$$

Where: ID_{CAN} – CAN message ID,
$Mask$ – *Mask* configuration parameter,
ID – *ID* configuration parameter,
& – Bitwise AND operator.

The *EID* box (*Extended ID* box) defines whether the CAN message ID is regular or extended.

All CAN ID mask filters run in parallel the same way as CAN ID range filters. It is sufficient to satisfy requirements of any active filter to send the CAN message to the Ethernet network.

If no active filters are defined, it is considered that the CAN ID mask filters are disabled, and do not participate in the message filtering process. In this case, *ID Mask Filters* are *Off* on the home page, and, if other filters are also disabled, all CAN input messages will be sent to the Ethernet network.

4 CONVERTER DIAGNOSTICS

The user can see a real-time diagnostic information on the *Diagnostics* page on the converter internal website. The connection to the converter embedded web server is described in the Converter Configuration section.

To see the *Diagnostics* page, Figure 14, the user should click on the *Diagnostics* link on the left side of the web page.

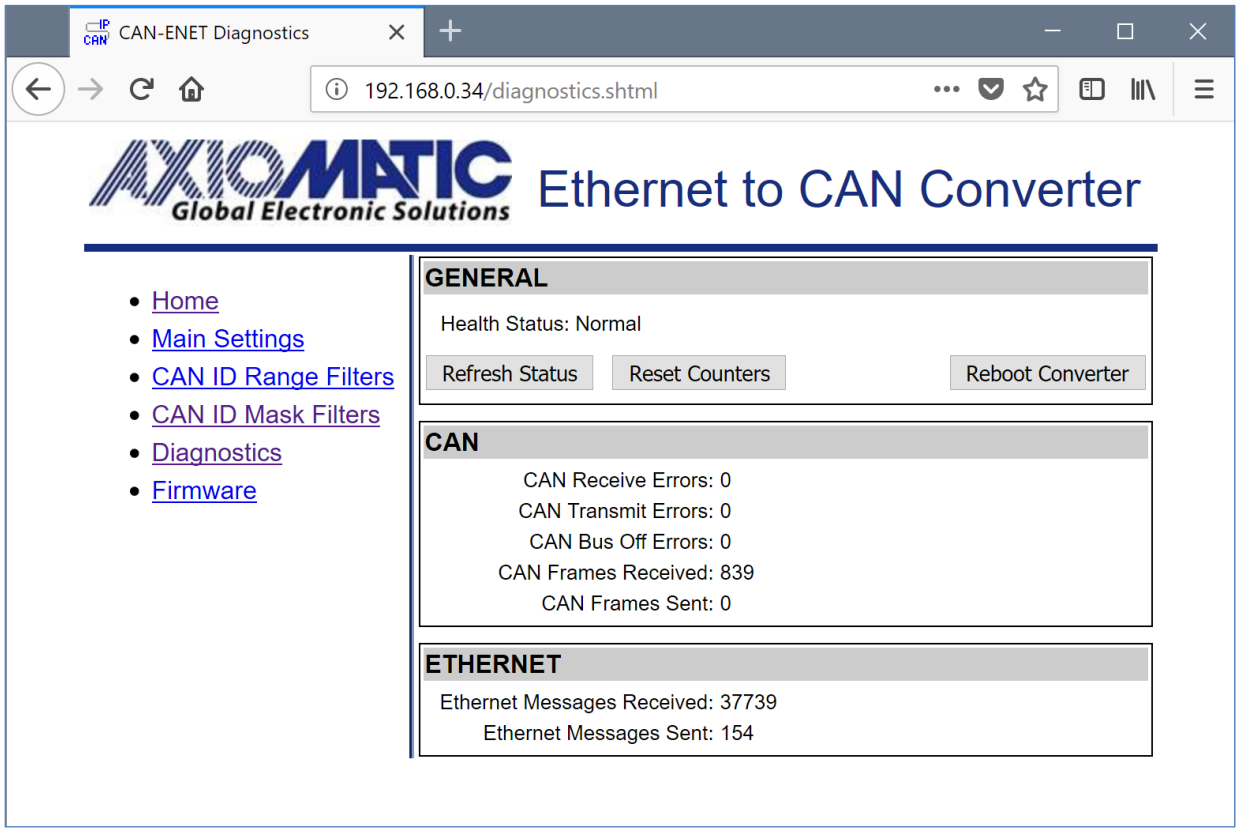


Figure 14. The Converter Diagnostics Page

The *Diagnostics* page shows the *Health Status* of the converter together with the *CAN* and *Ethernet* statistics.

The user can refresh the values on the page by pressing the *Refresh Status* button or reset the statistic counters by pressing *Reset Counters* button. The *Reboot Converter* button activates the converter rebooting.

The converters do not retain the diagnostic information. All information is lost when the power is shut down.

4.1 Health Status

The converter *Health Status* is an aggregated system run-time parameter calculated on the base of operational statuses of the major device hardware and software components.

The *Health Status* presents the overall operational status of the *CAN to Ethernet Converter*, based on the following rules, see Table 4.

Table 4. Health Status

Health Status	Condition
Error	*Error* is reported when at least one operational status is in the *Error* state.
Warning	*Warning* is reported when at least one operational status is in the *Warning* state and there are no operational statuses in the *Error* state.
Undefined	*Undefined* is reported when at least one operational status is in the *Undefined* state and there are no operational statuses in the *Error* or *Warning* state.
Normal	*Normal* is reported when all operational statuses are in the *Normal* state.

If the *Health Status* is different from *Normal*, the user will see a verbose message on the *Diagnostics* web page below the *Health Status* describing which operational status is causing a problem.

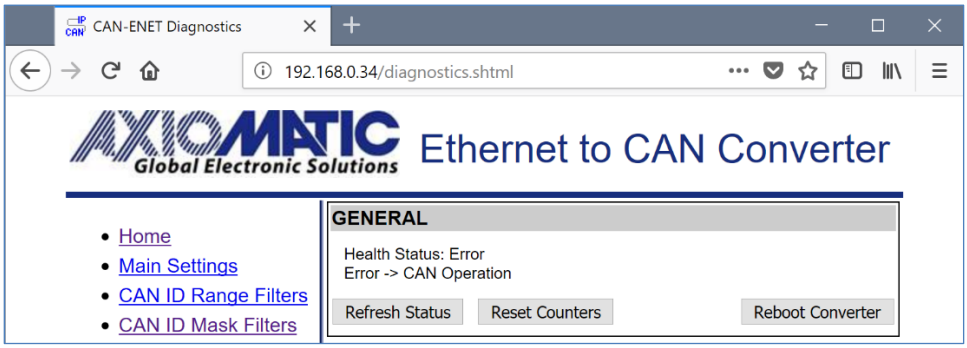


Figure 15. Health Status Message on CAN Error

In case, several operational statuses differ from *Normal*, all of them will be shown on the *Diagnostics* page.

4.2 Converter Rebooting

The user can reboot the converter, when necessary, using the *Reboot Converter* button.

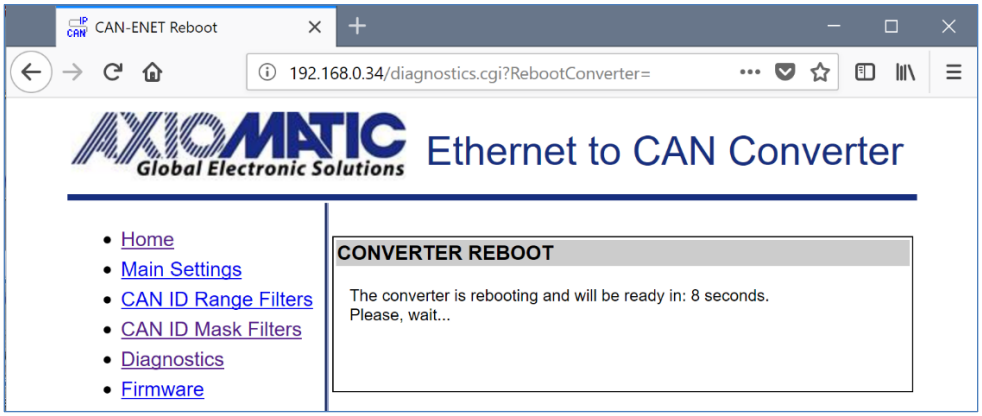


Figure 16. The Converter Reboot Screen

The rebooting operation takes 10 seconds. The user will see the *Reboot* screen with a countdown during this operation.

When rebooting is over, the *Diagnostics* page will be reloaded.

5 FIRMWARE UPDATE

The converter firmware can be updated through the website in the field

The update procedure is performed in two stages. First, the application firmware is uploaded into the converter internal flash. During this stage, the converter checks the firmware checksum and whether it can be programmed into the unit.

Then, upon the user confirmation, the firmware is programmed into the microcontroller and the unit is restarted. At the end of this process, the user should see the new firmware version number on the converter home page in the browser.

The details of the firmware update are provided below.

5.1 Uploading the New Firmware

To upload the new firmware, the user should activate the *Firmware Uploading* page, Figure 17, by clicking on the *Firmware* link on the left side of the web page.

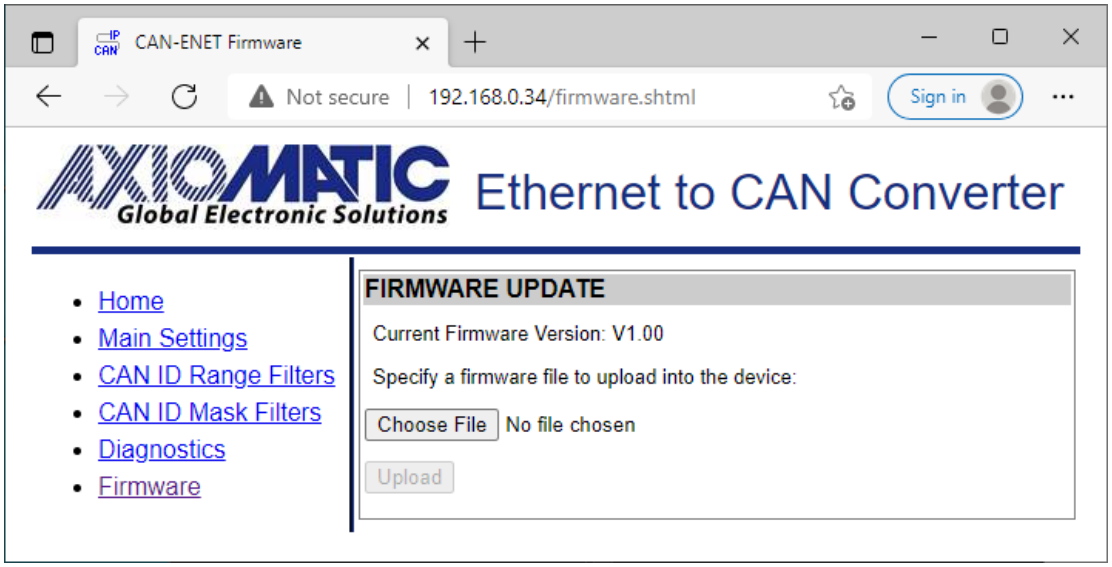


Figure 17. Firmware Uploading Page

Then the user selects the new firmware file using the *Browse…* button.

The firmware file is provided by Axiomatic in a proprietary binary format with extension: `.af`. The file name should have the following format: `AF-21153-X.XX.af`, where the `<X.XX>` field wildcard reflects the firmware version number. We will use `AF-21153-1.00.af` file for illustration of the firmware update process in this manual.

When the file is selected, the user should press the *Upload* button. The user will see the dynamic message: “Loading…” in the bottom of the screen and then, if everything is in order, the converter will switch automatically to the *Firmware Update* page.

5.2 Applying the New Firmware

On the *Firmware Update* page, the user will see the new firmware file information.

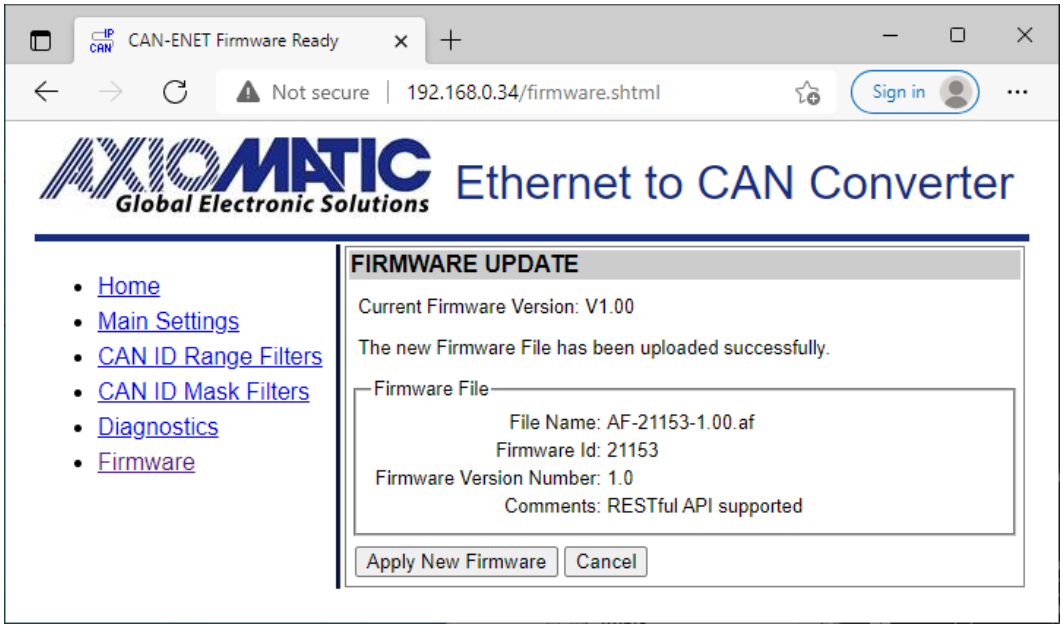


Figure 18. Firmware Update Page

From this point, the user can cancel the firmware update process and keep the old firmware or proceed with flashing the new firmware into the microcontroller by pressing the *Apply New Firmware* button.

When the user presses the *Apply New Firmware* button, the firmware update process is activated and the *Firmware Upload* page will show the countdown timer.

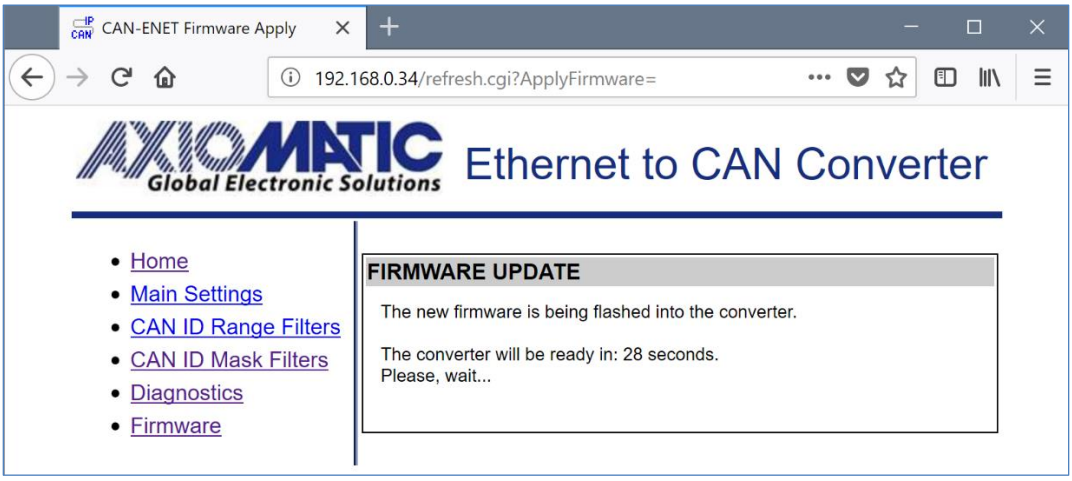


Figure 19. Firmware Update Countdown has been Started

The countdown timer is set for 30 seconds necessary to complete the flashing process and reboot the unit, after which the converter home page will be displayed. The user will see the new application firmware version number in the *Device Information* section on the converter home page.

6 CONVERTER DEPLOYMENT

There are two major approaches in using the Ethernet to CAN converter. One is to use the converter on its own as a CAN extender or a baud rate converter.

For example, a pair of coupled converters can synchronize two CAN networks. This example can be extended to several CAN networks running at different baud rate in various remote locations, connected using the Ethernet to CAN converters.

The second approach is to use the converter together with other IP devices that can directly communicate with the converter over the IP network. This approach requires writing a custom software for interfacing with the converter. Since the converter uses a proprietary communication protocol, Axiomatic provides the CAN-ENET Software Support Package (SSP), p/n AX140910, for interfacing with the converter. The SSP is downloadable from www.axiomatic.com, log-in section.

Most Axiomatic PC software tools support the Ethernet to CAN converter. They can connect to the CAN bus using the Ethernet to CAN converter the same way as they connect to the bus using the USB to CAN converter. The Axiomatic Electronic Assistant (EA) can communicate with the converter starting from version 5.11.82.0, and the CAN Assistant – Scope and CAN Assistant – Visual support the converter starting from version 3.0.0.

The use of the Ethernet to CAN converters for synchronizing CAN networks with or without baud rate conversion is described below. There is no need for custom software for this type of the converter deployment.

6.1 CAN Network Synchronization

To synchronize two remote CAN networks, the user can simply connect an Ethernet to CAN converter to each of the CAN network and configure the converters the way that they will talk with each other.

6.1.1 Hardware Setup

The converters should be connected through a local or global IP network, see: Figure 20 and Figure 21.

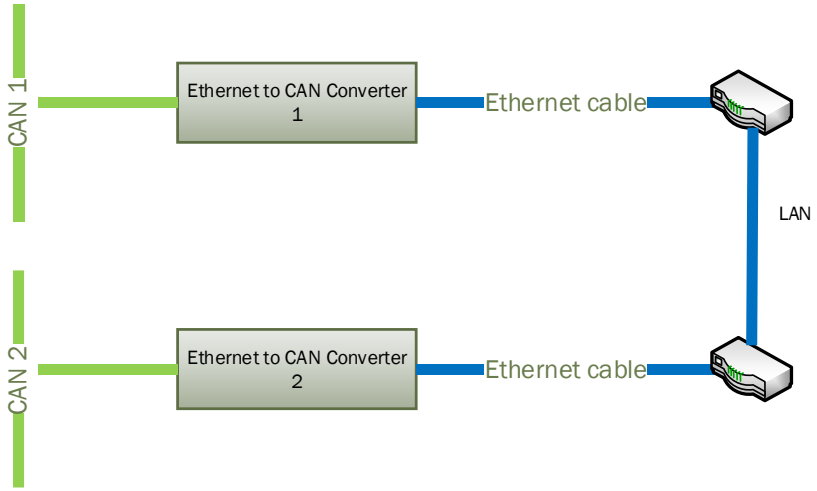


Figure 20. Local Connection through the LAN

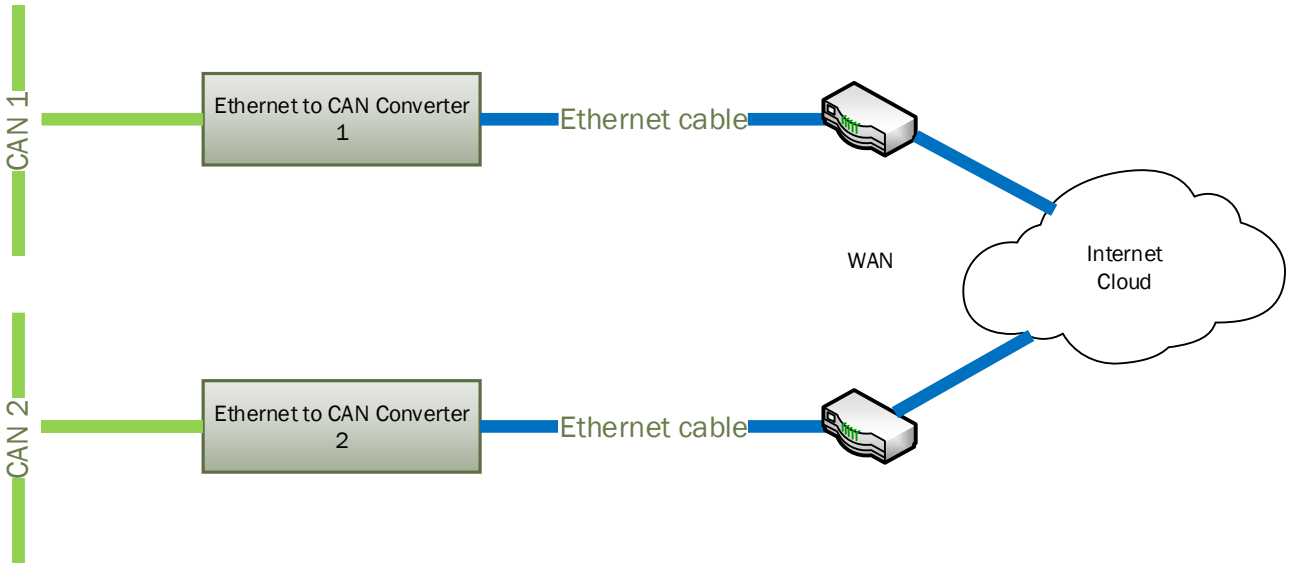


Figure 21. Global Internet Connection through the WAN

In the simplest scenario, two pre-configured converters can be connected by an Ethernet cable, see: Figure 22. Due to the Auto-MDIX feature, both: the straight and crossover cables can be used.

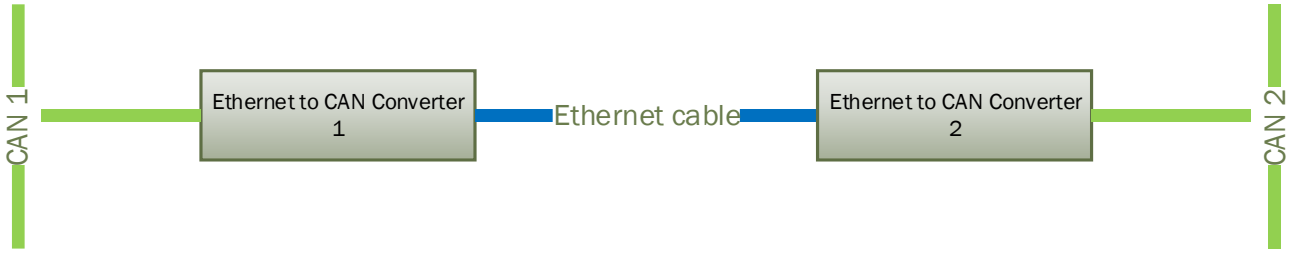


Figure 22. Simple Local Connection

The user can synchronize more than two CAN networks on LAN or WAN. There are practically no limits on the number of synchronized CAN networks, see daisy-chain connection in the Client Configuration section in Figure 25.

6.1.2 Converter Configuration

After the physical connection is established, the converters should be configured to exchange messages between each other. Since the converters support client/server communication model, one of the converters should be a server and the other one – a client.

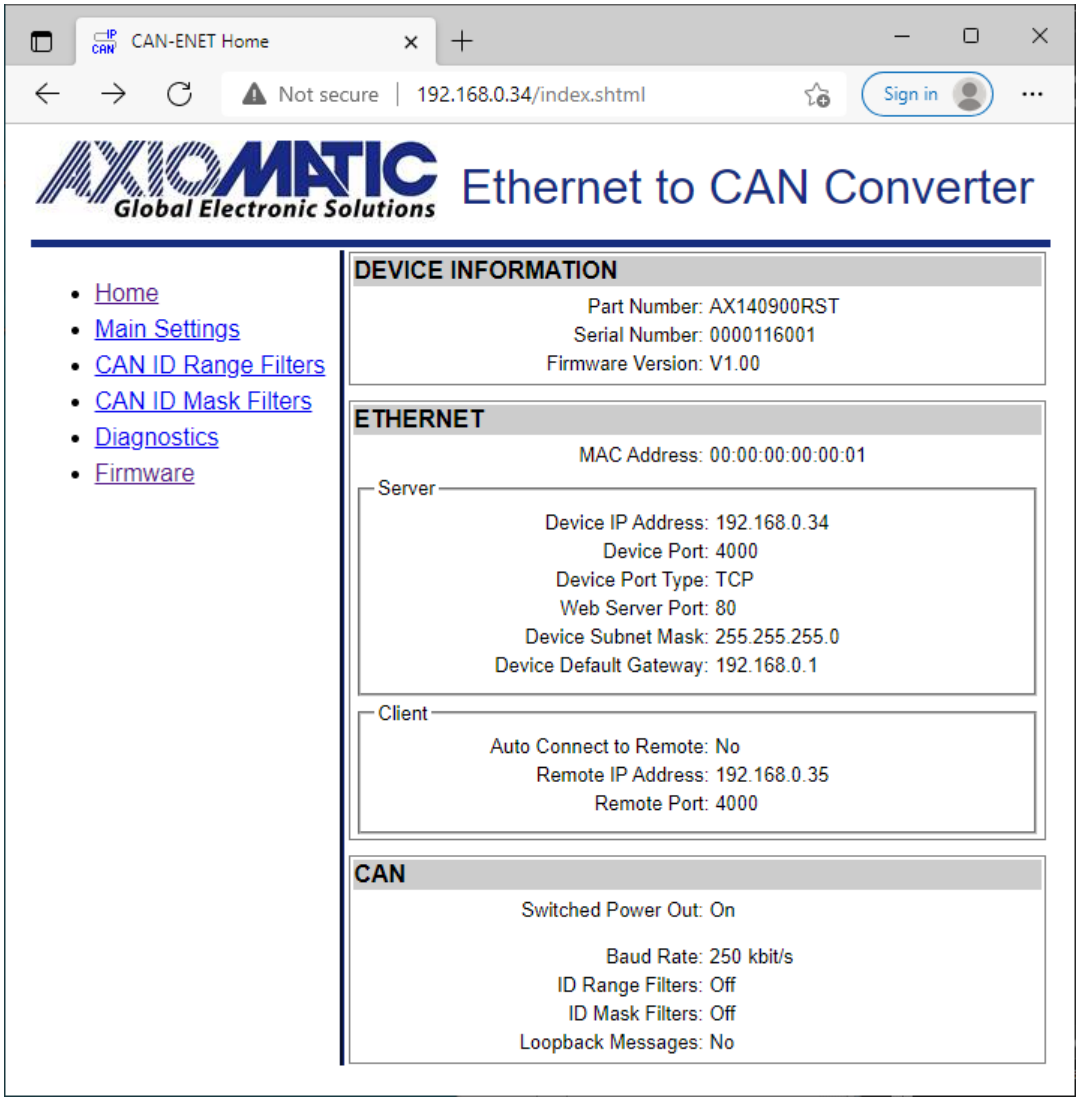
6.1.2.1 Server Configuration

To configure the converter as a server, first set: *Device IP Address*, *Device Port*, *Device Subnet Mask* and *Device Default Gateway* to the appropriate values received from your network administrator. Configure the *Device Port Type* to *UDP* or *TCP* depending on the required message reliability and acceptable delays. Use unreliable but fast UDP when speed is a priority or reliable but slow TCP when message reliability is more important than the speed.

Set *Auto Connect to Remote* to *No* to disable the client side of the converter. The *Remote IP Address* and *Remote Port* can be left untouched since they are not used by the converter when the client mode is disabled. They are grayed on the *Settings* page in this mode.

For the CAN network, configure the necessary *Baud Rate* and set *Loopback Messages* to *No*. The *Loopback Messages* configuration parameter must be set to *No* to avoid sending the same messages back and forth between converters, which will lead to network saturation and communication failure.

An example of the converter configuration as a server is presented in Figure 23.


CAN-ENET Home
Not secure | 192.168.0.34/index.shtml
Sign in
AXIOMATIC
Global Electronic Solutions
Ethernet to CAN Converter
Home
Main Settings
CAN ID Range Filters
CAN ID Mask Filters
Diagnostics
Firmware
DEVICE INFORMATION
Part Number: AX140900RST
Serial Number: 0000116001
Firmware Version: V1.00
ETHERNET
MAC Address: 00:00:00:00:00:01
Server
Device IP Address: 192.168.0.34
Device Port: 4000
Device Port Type: TCP
Web Server Port: 80
Device Subnet Mask: 255.255.255.0
Device Default Gateway: 192.168.0.1
Client
Auto Connect to Remote: No
Remote IP Address: 192.168.0.35
Remote Port: 4000
CAN
Switched Power Out: On
Baud Rate: 250 kbit/s
ID Range Filters: Off
ID Mask Filters: Off
Loopback Messages: No

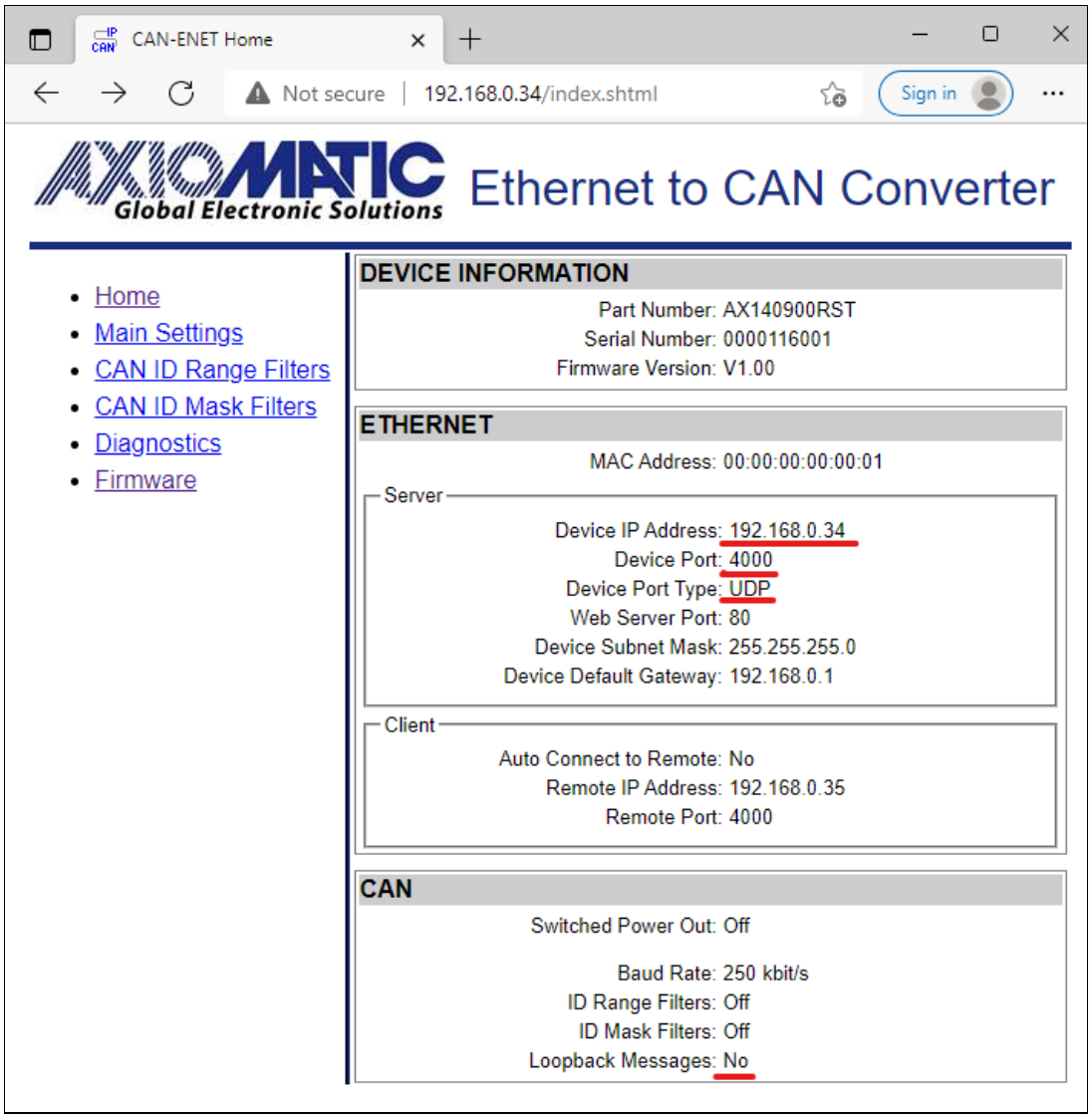


Figure 23. The Converter Server Configuration Example

6.1.2.2 Client Configuration

In the client configuration, the user should set *Device IP Address*, *Device Port*, *Device Subnet Mask* and *Device Default Gateway* to the appropriate values received from the network administrator the same way as with the server configuration. After that, the user should set the *Device Port Type, Remote IP Address* and *Remote Port* to match the settings of the converter in the server mode and activate the client mode by setting the *Auto Connect to Remote* to *Yes*.

The CAN network setup is done similarly to the server mode; the *Baud Rate* is set to the desired baud rate (not necessarily the same as on the server) and the *Loopback Messages* is set to *No*.

An example of the converter configuration in a client mode is presented in Figure 24.

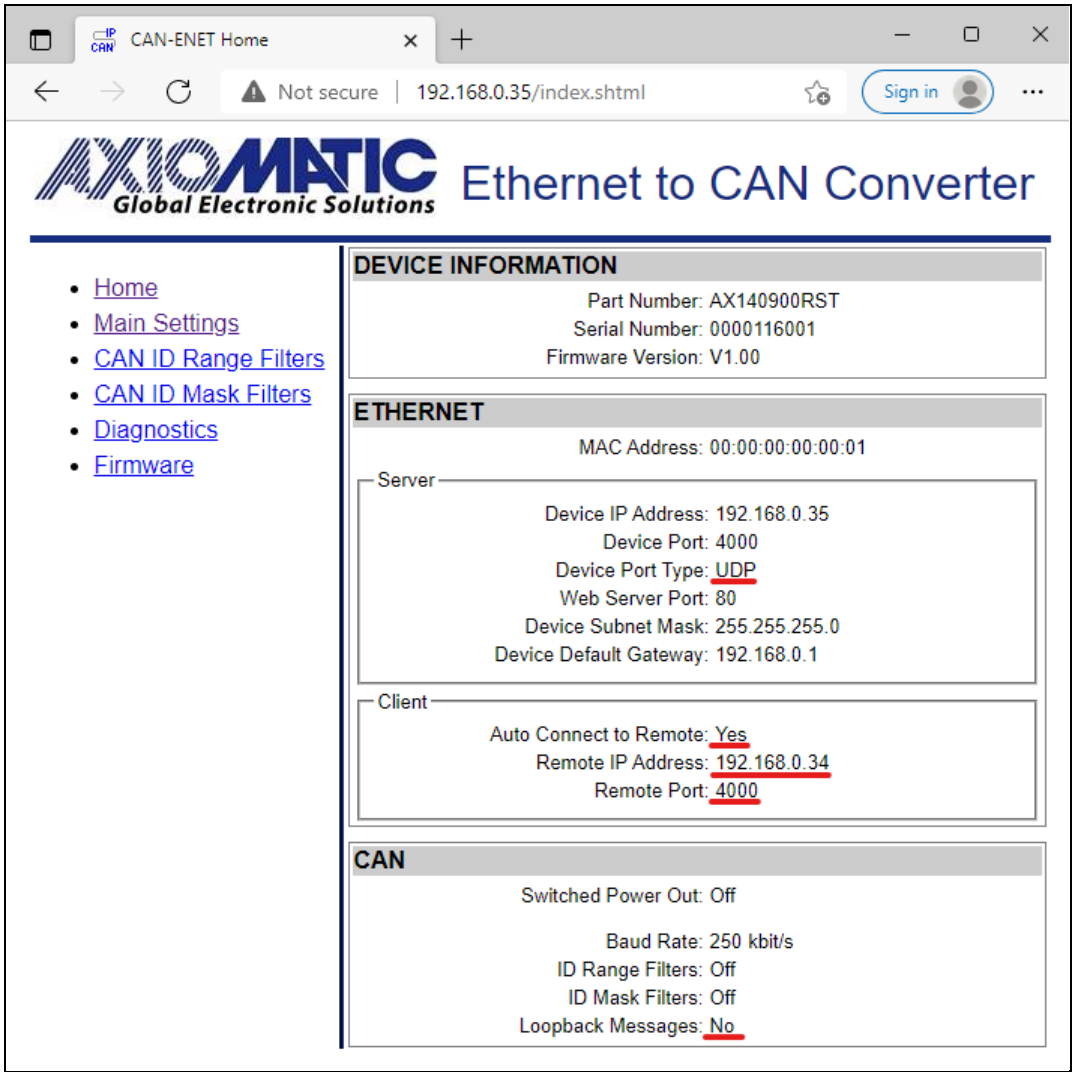


Figure 24. The Converter Client Configuration Example

The values of the *Device Port Type, Remote IP Address* and *Remote Port* of the client on Figure 24 are the same as the *Device Port Type*, *Device IP Address* and *Device Port* of the server in Figure 23.

Please note, that if the converters are connected over the internet, the *Remote IP Address* of the client will be a public IP address of the server, not the internal server IP address presented as the *Device IP Address* on Figure 23. The network administrator on the server side will be required to configure port forwarding to open internet access to the converter in the server mode.

When the converter is configured as a client, it will still act as a server accepting connections on the *Device Port* from other clients. This adds versatility to the converter configurations since the same converter can be used together with both: client and server communication nodes. As an example, the user can establish an unlimited number of daisy-chain client-server connections, see: Figure 25.

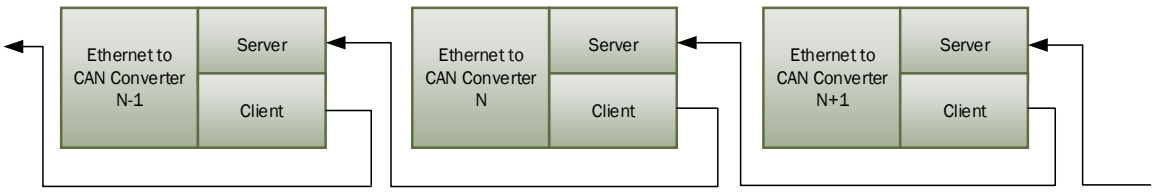


Figure 25. Daisy-Chain Converter Connection

7 CONVERTER DISCOVERY

In case the IP address and/or web server port is unknown or has been lost, the user can recover them using the Axiomatic `AxioDisc.exe` Windows console application. The application uses a proprietary discovery protocol and can recover IP locations of Axiomatic converters on a LAN. The `AxioDisc.exe` application is available upon request.

The application sends a UDP request to the global IP address `255.255.255.255`, port `35100` and waits for the responses from the converters located on the same local network. The converter discovery response includes the unit MAC Address, IP Address, web server port, device port, device port type, the converter part number and serial number.

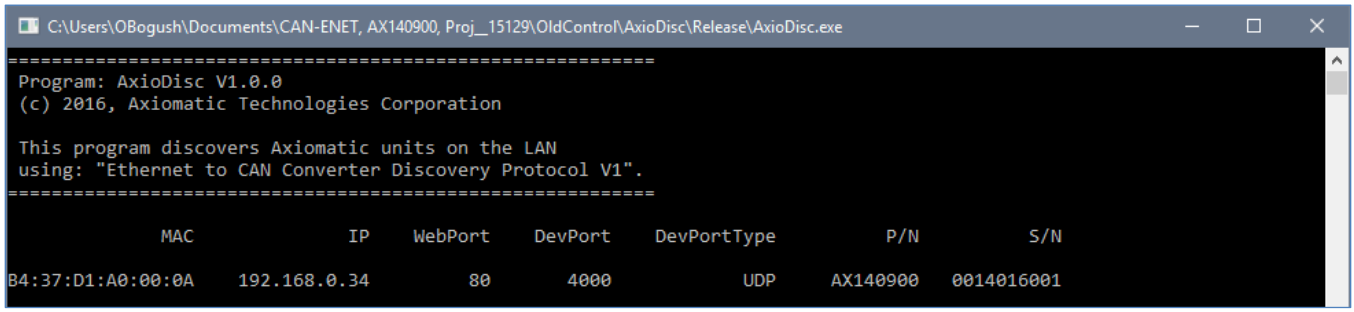


Figure 26. `AxioDisc.exe` Converter Discovery Application

The discovery protocol is also supported by the CAN-ENET Software Support Package, p/n AX140910, and can be used for network discovery and plug-and-play connection to Axiomatic Ethernet to CAN converters.

The `AxioDisc.exe` application can run on Windows starting from Win XP SP3. It was tested on Win XP SP3, Win 7 and Win 10. In case the application cannot find standard dlls, the Visual C++ Redistributable for Visual Studio 2015 x86 must be installed on the user's computer from the Microsoft website: https://www.microsoft.com/en-ca/download/details.aspx?id=48145

8 RESTful API Processing

A RESTful API is an architectural style for an application program interface (API) that uses HTTP requests to access and use data. The Converter's HTTP server uses the HTTP protocol at a predefined set of URLs and treats them as APIs. These URLs represent resources available at HTTP server's web page and have one or more methods that can be performed on them like GET, POST, PUT.

All HTTP API requests must have the encoding of *application/x-www-form-urlencoded*. Most APIs require one or more parameters. These parameters are passed as part of the message and have a rigid structure - parameter ID, followed by an equal sign, and then by the parameter value (i.e, *EthDeviceIpAddr*=192.168.0.34). Several parameters can be chained together with the ampersand[&] (i.e., *EthDeviceIpAddr*=192.168.0.34&*EthDevicePort*=4000). Blank spaces are not allowed. All parameters relevant to an API could be provided in the body of the same request.
For example, the following request sets the IP address to 192.168.1.34, port 4000 TCP:

```
PUT http://192.168.0.34/api/settings?EthDeviceIpAddr=192.168.1.34&EthDevicePort=4000&EthDevicePortType=TCP
```

A GET request is used to get a list of resources. If the resource is found on the server, then it returns HTTP response code 200 (OK) – along with the response body. The returned data is in JSON (JavaScript Object Notation) and can be accessed directly.
If the resource isn't found on the server, API returns HTTP response code 404 (NOT FOUND). Similarly, if it is determined that the GET request itself is not correctly formed then the server will return the HTTP response code 400 (BAD REQUEST).
A PUT request is used to update existing resources.
A POST request is used to upload a firmware.

8.1 Converter Settings

Converter's connection profile can be changed by posting PUT to */api/settings* the parameters listed in Table 5 (all are case sensitive) without limitations on number of parameters.

Table 5. RESTful API Parameters for Converter settings

API Parameter	Configuration Parameter	Range
DevPartNumber	Device Part Number	Read only
DevSerialNumber	Device Serial Number	Read only
DevFirmwareVersion	Device Firmware version	Read only
EthMACAddress	Device MAC Address	Read only
EthDeviceIpAddress	Device IP Address	Any IP address
EthDevicePort	Device Port	Any port value except the *Web Server Port* and the *Discovery Protocol Port*
EthDevicePortType	Device Port Type	{UDP, TCP}
EthWebServerPort	Web Server Port	Any port value except the *Device Port* and the *Discovery Protocol Port*
EthDeviceSubnetMask	Device Subnet Mask	Any IP address
EthDeviceDefaultGateway	Device Default Gateway	Any IP address
EthAutoConnectToRemote	Auto Connect to Remote	{0-No, 1-Yes}
EthRemoteIpAddress	Remote IP Address	Any IP address
EthRemotePort	Remote Port	Any port value

API Parameter	Configuration Parameter	Range
CANBaudRate	Baud Rate	{1000, 667, 500, 250, 125, 100, 83, 50, 20, 10}
CANIDRangeFilters	CAN Range Filters (Off/On)	Read only
CANIDMaskFilters	CAN Mask Filters (Off/On)	Read only
CANLoopbackState	Loopback Messages	{0-No, 1-Yes}
CANSwitchedPowerOutState	Switched Power Out	{0-Off, 1-On}

The Converter will send response in JSON-format :

```
{
"Status" : "success",
“Parsed parameters” :
“Changed parameters” :
}
```

In case of error or there were no new data to update the Converter will send error-response in JSON-format:

```
{
"Status" : "fail"
“Parsed parameters” :
“Changed parameters” :
“Fail parameter” :
“Error” :
}
```

A PUT to */api/settings_default* sets all Converter’s settings to default (see Table 2).

The Converter will use changed settings(including default settings) after reboot or PUT request to */api/settings?Apply*. After “Apply”-request the Converter replies with successful response:

```
PUT http://192.168.0.34/api/settings?Apply
{
"Status" : "success"
}
```

or fail response with error “No Data to Write” if it didn’t receive any new data.
Information on the existing Converter’s profile accessed by GET request to */api/settings*.

```
GET http://192.168.0.34/api/settings
{
"DevPartNumber" : "AX140900RST",
"DevSerialNumber" : "0000116001",
"DevFirmwareVersion" : "V1.00",
"EthMACAddress" : "00:00:00:00:00:01",
"EthDeviceIpAddress" : "192.168.0.34",
"EthDevicePort" : "4000",
"EthDevicePortType" : "UDP",
"EthWebServerPort" : "80",
"EthDeviceSubnetMask" : "255.255.255.0",
"EthDeviceDefaultGateway" : "192.168.0.1",
"EthAutoConnectToRemote" : "No",
"EthRemoteIpAddress" : "192.168.0.35",
"EthRemotePort" : "4000",
"CANBaudRate" : "250",
"CANIDRangeFilters" : "Off",
"CANIDMaskFilters" : "Off",
"CANLoopbackState" : "No",
"CANSwitchedPowerOutState" : "Off"
}
```

For example, the following request sets the Loopback to ON, the Subnet mask to 255.255.255.0, the Switched Power Out to Off, the Device Port to 4111, the Device Port Type to TCP:

```
PUT
http://192.168.0.34/api/settings?CANLoopbackState=1&EthDeviceSubnetMask=255.255.255.0&CANSwitchedPowerOutState=0&EthDevicePort=4111&EthDevicePortType=TCP
{
"Status" : "success",
"Parsed parameters" : "5",
"Changed parameters" : "3"
}
```

Then command to apply changes :

```
PUT http://192.168.0.34/api/settings?Apply
{
"Status" : "success"
}
```

8.2 CAN Filters

CAN Range Filters profile can be changed by posting PUT to */api/CANRangeFilters* the parameters listed in Table 6 (all are case sensitive) without limitations on number of parameters.

Table 6. CAN Range Filters RESTful API Parameters

API Parameter	Configuration Parameter	Range
CANRangeFilter[1..5]OnOff	Activate range for filter [1..5]	{0-Off, 1-On}
CANRangeFilter[1..5]From	*From* configuration parameter	hex-data
CANRangeFilter[1..5]To	*To* configuration parameter	hex-data
CANRangeFilter[1..5]EID	Extended ID	{0-Off, 1-On}

Information on the existing CAN Range Filters - GET to */api/CANRangeFilters*.

```
GET http://192.168.0.34/api/CANRangeFilters
{
"CANRangeFilter1OnOff" : "Off",
"CANRangeFilter1From" : "0x0",
"CANRangeFilter1To" : "0x0",
"CANRangeFilter1EID" : "Off",
……..
}
```

CAN Mask Filters profile can be changed by posting PUT to */api/CANMaskFilters* the parameters listed in Table 7 (all are case sensitive) without limitations on number of parameters.

Table 7. CAN Mask Filters RESTful API Parameters

API Parameter	Configuration Parameter	Range
CANMaskFilter[1..5]OnOff	Activate mask for filter [1..5]	{0-Off, 1-On}
CANMaskFilter[1..5]From	*From* configuration parameter	hex-data
CANMaskFilter[1..5]To	*To* configuration parameter	hex-data
CANMaskFilter[1..5]EID	Extended ID	{0-Off, 1-On}

Information on the existing CAN Mask Filters - GET to */api/CANMaskFilters*.

```
GET http://192.168.0.34/api/CANMaskFilters
{
"CANMaskFilter1OnOff" : "Off",
"CANMaskFilter1Mask" : "0x0",
"CANMaskFilter1ID" : "0x0",
"CANMaskFilter1EID" : "Off",
……..
}
```

The Converter sends response in JSON-format :

```
{
"Status" : "success",
“Parsed parameters” :
“Changed parameters” :
}
```

In case of error or there were no new data to update the Converter will send error-response in JSON-format:

```
{
"Status" : "fail"
“Parsed parameters” :
“Changed parameters” :
“Fail parameter” :
“Error” :
}
```

A PUT to */api/CANFilter_default* clears all CAN Filter configuration parameters.

Changed parameters became active immediately.

For example, the following request sets the 2nd Range Filter’s EID to ON:

```
PUT http://192.168.0.34/api/CANRangeFilters?CANRangeFilter2EID=1
{
"Status" : "success",
"Parsed parameters" : "1",
"Changed parameters" : "1"
}
```

8.3 Converter Diagnostics

A real-time diagnostic information can be seen by posting GET to */api/diagnostics*.

```
GET http://192.168.0.34/api/diagnostics

{
"GenHealthStatus" : "Normal",
"Can Receive Errors" : "0",
"CANTransmitErrors" : "0",
"CANBusOffErrors" : "0",
"CANFramesReceived" : "0",
"CANFramesSent" : "0",
"EthMessagesReceived" : "136",
"EthMessagesSent" : "6"
}
```

A PUT to */api/diagnostics?ResetDiagnostics* resets diagnostics counters.
A PUT to */api/diagnostics?RebootConverter* reboots the Converter.

For example, the following request resets all diagnostics:

```
PUT http://192.168.0.34/api/diagnostics?ResetDiagnostics
{
"Status" : "success"
}
```

The following request reboots the converter:

```
PUT http://192.168.0.34/api/diagnostics?RebootConverter
{
"Status" : "success"
}
```

8.4 Firmware Update

The update procedure can be performed by HTTP POST request with MIME type multipart/form-data to */api/firmware_update*. After uploading the Converter checks the firmware checksum and whether it can be programmed into the unit, then the firmware is programmed into the microcontroller and the unit is restarted.
The Converter sends success-response with information about uploaded firmware in JSON-format :

```
{
"Status" : "success, converter restarting",
"DevSerialNumber" : "AF-21153-1.00.af",
"FlashFileName" : "AF-21153-1.00.af",
"FlashFirmwareId" : "21153",
"FlashFirmwareVerNum" : "1.0",
"FlashFirmwareComments" : "RESTful API supported"
}
```

In case of failure the Converter will send error-response in JSON-format:

```
{
"Status" : "fail"
“File name” :
“Error code” :
“Detail” :
}
```

The update procedure can also be performed in two stages:

1) Send HTTP POST request to */api/firmware*. During this stage the Converter checks the firmware and uploads into the unit.

```
{
"Status" : "success ",
"DevSerialNumber" : "AF-21153-1.00.af",
"FlashFileName" : "AF-21153-1.00.af",
"FlashFirmwareId" : "21153",
"FlashFirmwareVerNum" : "1.0",
"FlashFirmwareComments" : "RESTful API supported"
}
```

2) Send HTTP PUT request to */api/fimware?Apply*. During this stage the firmware is programmed into the microcontroller and the unit is restarted.

After stage 1, a GET to */api/fimware?UploadedFileInfo* will respond with information about uploaded firmware. A PUT to */api/fimware?Cancel* will clean uploaded firmware from the Controller’s memory.

9 TECHNICAL SPECIFICATIONS

9.1 Power Supply

9.1.1 Input

Power supply input is located on the Ethernet connector. The power supply uses automotive battery power. It is not compatible with the PoE (Power over Ethernet) IEEE 802.3 standard.

Table 8. Power Supply Input

Parameter	Value	Remarks
Supply Voltage	9…36 VDC	12V, 24V – nominal
Power Consumption	2W	Maximum at 12V
Protection	Under Voltage Shutdown Over Voltage Shutdown Reverse Polarity Transients	< 6 V > 37V 12V Load Dump

9.1.2 Output

Power supply output is located on the CAN connector.

Table 9. Power Supply Output

Parameter	Value	Remarks
Voltage Output	9…36 VDC	Pass-through power from the power supply input.
Current Output	0.7A	Maximum pass-through current.
Voltage Drop	1V	Maximum
Protection	Overcurrent Short to Battery Short to Ground	> 1A

9.2 Ethernet

Table 10. Ethernet Parameters

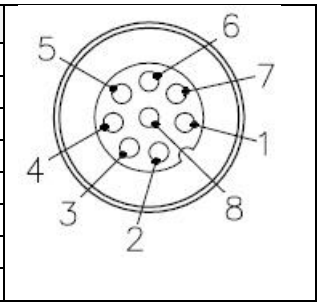
Parameter	Value	Remarks
Number of Ports	1	
Port Type	10BASE-T, 100BASE-TX	Auto-configuration and full-duplex supported.
MDIX	Auto-MDIX	Auto-crossover to eliminate cabling mismatch.
LED Indicators	Speed/Activity	
Protocols	Ethernet IEEE 802.3, IP, ICMP, ARP, UDP, TCP, HTTP, Proprietary	CAN messages are transmitted using a proprietary application protocol running on top of the user selectable UDP or TCP transport protocol. The internal web server uses HTTP protocol. The unit supports a proprietary discovery protocol.
Server Mode	Up to 10 bi-directional simultaneous connections	Up to 9 connections, if the Client mode is enabled.
Client Mode	1 remote bi-directional connection	Auto-connect to a remote server, if connection is dropped or temporary unavailable. Client mode can be disabled.
Web server	Provided	Always enabled for converter configuration and diagnostics.
Internal Diagnostics	Health Status	Internal health status of the converter is transmitted in heartbeat messages. It is also available from the web server.

9.2.1 Ethernet Connector

M12 socket, 8-pin, A-coded, female connector, Phoenix Contact, P/N: 1441817.

Table 11. Ethernet Connector Pinout

PIN #	Description
1	BAT + (9-36V)
2	BAT – (GND)
3	BAT – (GND)
4	ETH_TX –
5	ETH_RX +
6	ETH_TX +
7	BAT + (9-36V)
8	ETH_RX –



Use A-coded mating connectors compliant with IEC 61076-2-101:2012.

The AX070531 Ethernet and Power Cable - 1.7m (5.5 ft.), 8-pin M12 A-coded, Unterminated Leads, Ethernet Jack, can be used for experimenting. The cable is rated -40…+75 °C.

9.3 CAN

Table 12. CAN Parameters

Parameter	Value	Remarks
Number of Ports	1	
Port Type	High Speed, ISO 11898-2 compatible	Twisted pair, up to 1 Mbit/s. Shield connection is provided if shielded cable is used.
Baud Rate	1000, 666.6(6), 500, 250, 125, 100, 83.3(3), 50, 20, 10[1]	[kbit/s]. Programmable through web interface.
Protocol	CAN Bosch 2.0A and B	Data Frames and Remote Frames with Standard and Extended IDs are supported. CAN ID range and mask filtering is provided.

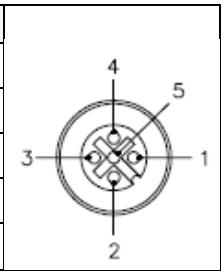
[1] *Support for 666.6(6) kbit/s is absent in firmware versions 1.xx…4.xx.*

CAN port does not contain 120 Ohm termination resistor.

9.3.1 CAN Connector

M12 socket, 5-pin, A-coded, female connector, Phoenix Contact, P/N: 1441778.

PIN #	Description
1	CAN_SHIELD
2	POUT + (Switch Output)
3	POUT – (GND)
4	CAN_H
5	CAN_L



Use mating A-coded connectors compliant with IEC 61076-2-101:2012.

The AX070532 CAN Cable - 1.5 m (5 ft.), 5-pin M12 A-coded, Unterminated Leads, can be used for experimenting. The cable is rated -40…+105 °C.

9.4 General Specifications

Table 13. General Specifications

Parameter	Value	Remarks
Operating Temperature	-40…+85 °C	Industrial temperature range
Storage Temperature	-40…+85 °C	
Environmental Protection	IP67	IEC 60529
Vibration	Sine sweep, 5-200 Hz, 8.9G peak, 2.5 hr (15 sweeps), each axis. Random, 10-1014 Hz, 6.86 Grms, 5.0 hr, each axis.	Custom profile
Shock	50G peak, 5 shocks, each axis	Custom profile
Size	4.19in x1.82in x1.32in (107mm x 47mm x 34 mm)	See dimensional drawing
Weight	0.15 lb (0.068 kg)	
Compliance	RoHS Directive	

Table 14. Electromagnetic Compatibility (EMC)

Standard	Description
EN 13309: 2010 ISO 13766-1:2018	Construction Machinery- Electromagnetic Compatibility of Machines with Internal Electrical Power Supply. Earth-moving and building construction machinery - Electromagnetic compatibility (EMC) of machines with internal electrical power supply - Part 1: General EMC requirements under typical electromagnetic environmental conditions
EN61000-6-4:2005	Emission Standard for Industrial Environments
EN61000-6-2:2007	Generic Standards – Immunity for Industrial Environments
EN61000-4-2	ESD, Level 4 (15 kV air/8 kV contact) at Criteria B
EN61000-4-4	EFT/burst Immunity, Test Level 1 (500V power/250V signal lines) at Performance Criteria B
EN61000-4-5	Surge Immunity, Test Level 1 (500V) at Performance Criteria B

9.5 Accessories

Table 15. Accessories

Axiomatic P/N	Description
AX140910	Software Support Package (SSP). Downloadable from www.axiomatic.com, log-in section.
AX070531	AX070531 Ethernet and Power Cable - 1.7m (5.5 ft.), 8-pin M12 A-coded, Unterminated Leads, Ethernet Jack.
AX070532	CAN Cable - 1.5 m (5 ft.), 5-pin M12 A-coded, Unterminated Leads.
AX140900RSTK	Service tool kit, contains: • AX140900RST Ethernet-CAN Converter, RESTful; • AX070531 Ethernet and Power Cable - 1.7m (5.5 ft.), 8-pin M12 A-coded, Unterminated Leads, Ethernet Jack; • AX070532 CAN Cable - 1.5 m (5 ft.), 5-pin M12 A-coded, Unterminated Leads.

9.6 Housing

Injection molded enclosure and cover. Material: PA66, 30% glass fiber reinforced, flame retardant UL 94 V-0. Ultrasonically welded. For dimensional drawing, see Figure 27.

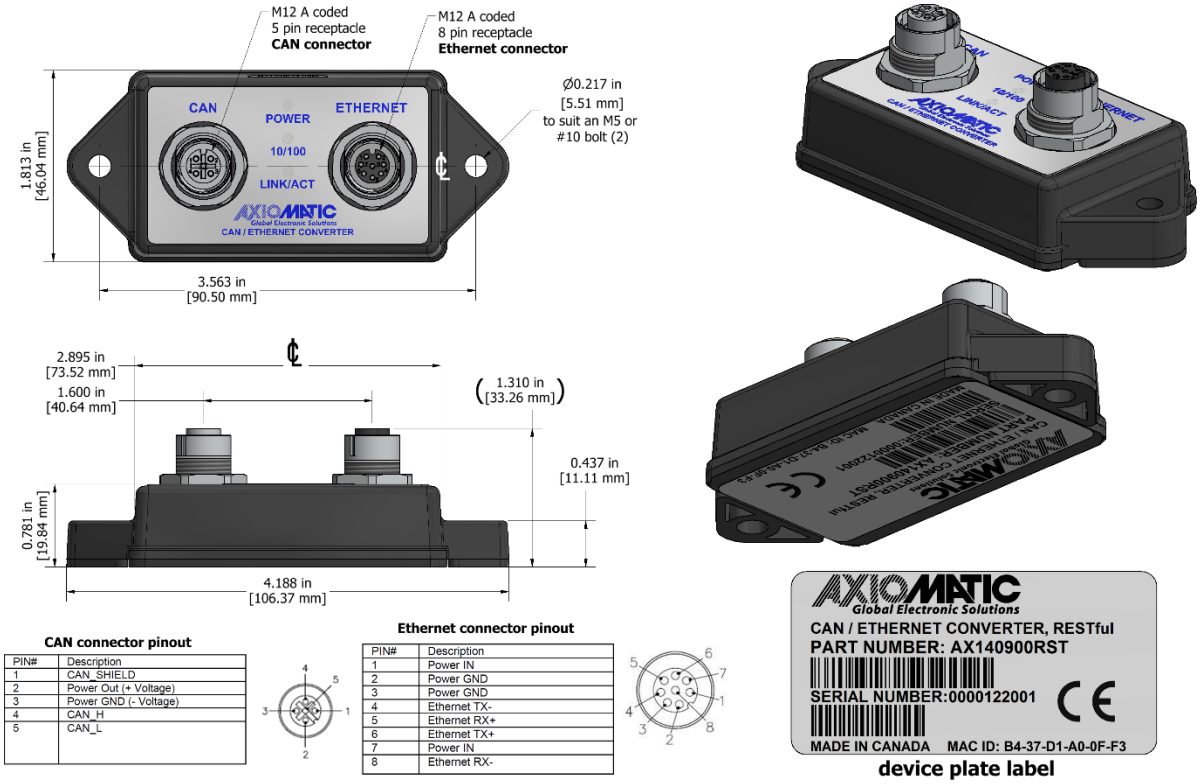


Figure 27. Dimensional Drawing

10 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 Ethernet to CAN Converter firmware.

Table 16. Third Party Software License Notices

Third Party Software	License Notice/Terms
STMicroelectronics microcontroller support software	COPYRIGHT(c) 2015 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. COPYRIGHT(c) 2014 STMicroelectronics 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 V9.0.0	Copyright (C) 2016 Real Time Engineers Ltd. All rights reserved FreeRTOS is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License (version 2) as published by the Free Software Foundation >>>> AND MODIFIED BY <<<< the FreeRTOS exception. ********** >>! NOTE: The modification to the GPL is included to allow you to !<< >>! distribute a combined work that includes FreeRTOS without being !<< >>! obliged to provide the source code for proprietary components !<< >>! outside of the FreeRTOS kernel. !<< **********

	FreeRTOS is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. Full license text is available on the following link: http://www.freertos.org/a00114.html
LwIP v2.0.2	Copyright (c) 2001-2004 Swedish Institute of Computer Science. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. 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. 3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``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 AUTHOR 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. The license is available at: http://lwip.wikia.com/wiki/License

11 VERSION HISTORY

User Manual Version	Firmware version	Date	Author	Modifications
1	1.xx	January 25, 2022	Viktor Rogachov	Added RESTful API support. Based on UMAX140900 V6
-	-	January 26, 2022	Amanda Wilkins	Updated p/n's in Technical spec accessories Updated compliance in Technical spec
1.1	-	September 12, 2023	Kiril Mojsov	Performed Legacy Updates

OUR PRODUCTS

AC/DC Power Supplies

Actuator Controls/Interfaces

Automotive Ethernet Interfaces

Battery Chargers

CAN Controls, Routers, Repeaters

CAN/WiFi, CAN/Bluetooth, Routers

Current/Voltage/PWM Converters

DC/DC Power Converters

Engine Temperature Scanners

Ethernet/CAN Converters, Gateways, Switches

Fan Drive Controllers

Gateways, CAN/Modbus, RS-232

Gyroscopes, Inclinometers

Hydraulic Valve Controllers

Inclinometers, Triaxial

I/O Controls

LVDT Signal Converters

Machine Controls

Modbus, RS-422, RS-485 Controls

Motor Controls, Inverters

Power Supplies, DC/DC, AC/DC

PWM Signal Converters/Isolators

Resolver Signal Conditioners

Service Tools

Signal Conditioners, Converters

Strain Gauge CAN Controls

Surge Suppressors

OUR COMPANY

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

QUALITY DESIGN AND MANUFACTURING

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

WARRANTY, APPLICATION APPROVALS/LIMITATIONS

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

COMPLIANCE

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

SAFE USE

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

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

SERVICE

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

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

DISPOSAL

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

CONTACTS

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

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