



# DAKOTA SOFT

## FUNCTIONAL MANUAL FOR THE TRANSLATOR X2™ USB/CAN/SERIAL CONVERTER & DATA LOGGER

**VERSION: 1.15.00**

December 15, 2017



Copyright © 2010 - 2017 DakotaSoft, Inc.

All Rights Reserved

# 1.Contents

<b>COPYRIGHT © 2010 - 2017 DAKOTASOFT, INC. ....</b>	<b>1</b>
<b>ALL RIGHTS RESERVED .....</b>	<b>1</b>
<b>1. CONTENTS .....</b>	<b>2</b>
<b>2. REVISION HISTORY .....</b>	<b>4</b>
<b>3. CONTACT INFORMATION.....</b>	<b>6</b>
<b>4. SCOPE AND BACKGROUND INFORMATION.....</b>	<b>7</b>
<b>5. INSTALLATION INSTRUCTIONS.....</b>	<b>7</b>
5.1 PARTS SHIPPED .....	7
5.2 SOFTWARE INSTALLATION.....	7
<b>6. OPERATIONAL OVERVIEW .....</b>	<b>7</b>
<b>7. ENCLOSURE.....</b>	<b>8</b>
<b>8. HARDWARE SPECIFICATIONS.....</b>	<b>8</b>
8.1 CONNECTORS.....	8
8.1.1 CAN Port 1– DB9 Pinout.....	8
8.1.2 CAN Port 2 & Power – DB9 Pinout .....	9
8.1.3 RS-232 – DB9 Pinout.....	9
8.1.4 USB Port – Female B.....	9
8.1.5 USB Port – Female A.....	10
8.2 POWER.....	10
8.3 PROCESSOR/MEMORY.....	10
8.4 ENVIRONMENTAL .....	10
<b>9. COMMUNICATIONS PORTS .....</b>	<b>10</b>
9.1 CAN PORTS .....	11
9.2 1 FULL SPEED USB DEVICE PORT .....	11
9.3 1 FULL SPEED USB HOST PORT.....	11
9.4 1 RS232 PORT .....	11
<b>10. USB HOST PORT.....</b>	<b>11</b>
<b>11. USB DEVICE PORT .....</b>	<b>12</b>
<b>12. CAN BUS MONITOR .....</b>	<b>12</b>
12.1 MAIN WINDOW.....	12
12.1.1 CAN 1 & CAN 2 Port Tabs .....	13
12.1.2 Date/Time Tab.....	16
12.1.3 Filter Config Tab .....	17
12.1.4 Log Setup Tab .....	19
12.1.5 Log Playback Tab .....	22
12.1.6 Message Converter Tab .....	24
12.1.7 RS-232 Tab.....	28
12.1.8 Upload Software Tab .....	30

12.2	FILE MENU .....	32
12.3	EDIT MENU.....	33
12.4	VIEW MENU.....	33
12.5	CONFIGURE BUS MENU .....	33
12.6	UNITS MENU.....	34
12.7	HELP MENU .....	35
<b>13.</b>	<b>USB API.....</b>	<b>36</b>
13.1	USB VENDER ID & PRODUCT ID.....	36
13.2	OVERALL MESSAGE STRUCTURE.....	36
13.2.1	<i>CRC16 Code and Lookup Table.....</i>	36
13.3	RX CAN MESSAGE.....	37
13.4	TX CAN MESSAGE .....	37
13.5	RESET USB PORT MESSAGE.....	38
13.6	RESET CAN PORTS MESSAGE.....	38
13.7	SET HARDWARE FILTERS MESSAGE.....	38
13.8	CHANGE CYCLIC TX MESSAGE.....	39
13.9	REMOVE CYCLIC TX MESSAGE.....	39
13.10	CHANGE BAUD RATE MESSAGE .....	39
13.11	GET BAUD RATE MESSAGE .....	40
13.12	INITIALIZE LOG PAYBACK TIMER MESSAGE.....	41
13.13	GET MESSAGE CONVERTER DATA MESSAGE .....	41
13.14	CHANGE MESSAGE CONVERTER DATA MESSAGE .....	42
13.15	GET TRANSLATOR TIMER TICK.....	43
13.16	CLEAR USB Tx BUFFER .....	43
13.17	SET TRANSLATOR TIMER TICK .....	44
13.18	GET TRANSLATOR SETTINGS MESSAGE.....	44
13.19	SAVE FILTERS MESSAGE.....	46
13.20	SAVE SD CARD LOG FILE SETTINGS MESSAGE .....	47
13.21	READY TO RX CAN MESSAGES FLAG MESSAGE.....	47
13.22	CHANGE GATEWAY SETTINGS MESSAGE.....	47
13.23	FORMAT SD CARD MESSAGE .....	48
13.24	GET REVISION INFORMATION MESSAGE.....	48

## 2.Revision History

---

### Revision 1.00.03

Initial Release

Date: Dec. 28, 2014

Author: Ty Johnson

### Revision 1.01.00

Date: Feb. 17, 2015

Author: Ty Johnson

Fixed a bug where the log playback feature would not playback a log that had messages from both CAN ports to their respective CAN ports.

### Revision 1.02.00

Skipped this revision number.

### Revision 1.03.00

Date: April 23, 2015

Author: Ty Johnson

Fixed a bug where the Translator would sometimes generate a hard fault if the RS-232 port is receiving messages and a thumb drive is inserted into the USB host port to copy files off the SD Card.

### Revision 1.04.00

Skipped this revision number.

### Revision 1.05.00

Date: Oct 28, 2015

Author: Ty Johnson

- Removed the mass storage device feature from Translator x2.

- Fixed a bug in the UART string receive function where the string start point was not always correct for the second string placed in the buffer.

- Fixed a bug in the SD Card initialization software where a register was not given enough time to accept its change before another register that depended on the value of the first register was changed.

- Replaced the Keil FAT with new FAT.

### Revision 1.06.00

Date: Dec. 3, 2015

Author: Ty Johnson

- Fixed some bugs with the download of log files to the thumb drive. These bugs happened when the thumb drive ran out of memory. One bug caused an endless loop. The other bug caused the sd card to be erased without saving any data to the thumb drive.

- Increased the download time of log files to the sd card using various methods. Note: The transfer rate from SD Card to thumb drive is ~740k/s.

### Revision 1.07.00

Date: Dec. 17, 2015

Author: Ty Johnson

- Removed the ability of the CAN Bus Monitor software to format the SD Card.

### Revision 1.08.00

Date: Jan. 6, 2016

Author: Ty Johnson

- Added an auto baud feature to the RS-232 port.

- Added a counter filter to the CAN port received messages logging.

### Revision 1.09.00

Date: Jan. 15, 2016

Author: Ty Johnson

- Added port number 3 to the RS-232 entry in the log files.

## Revision 1.10.00

Date: Jan. 30, 2016

Author: Ty Johnson

- Fixed a bug in the log file where there would be 7 extra random characters in the first log entry.

## Revision 1.11.00

Date: Apr. 12, 2016

Author: Ty Johnson

- Fixed a bug where the transport protocol connection mode messaging would not work on CAN port 2.

## Revision 1.12.00

Date: Apr. 12, 2016

Author: Ty Johnson

- Fixed a bug where only the first filter would work for the CAN filters. This error was introduced in version 1.08.00.

## Revision 1.13.00

Date: July 8, 2016

Author: Ty Johnson

- Fixed a bug where the terminating resistors on both CAN ports would remain on regardless of the CAN Bus Monitor x2 setting.

## Revision 1.14.00

Date: Oct 12, 2016

Author: Ty Johnson

- Fixed a bug where not all memory sticks would work on the USB host port.

## Revision 1.15.00

Date: Dec 15, 2016

Author: Ty Johnson

- Added the Tx Response check box. When check the software will respond to the ISO Request Message with the message that is requested if that message is loaded in the Tx Messages section of the CAN Port tab.

### 3. Contact Information

---

DakotaSoft, Inc.  
1459 CR5100  
Coffeyville, KS 67337  
620.251.0067

Sales:  
[sales@dakotasoftinc.com](mailto:sales@dakotasoftinc.com)

Customer Service:  
[support@dakotasoftinc.com](mailto:support@dakotasoftinc.com)

## 4. Scope and Background Information

---

The purpose of this document is to provide documentation for the operation of the Translator x2™ USB/CANSerial converter and data logger device. Within this document you will find information concerning the Translator x2™ hardware connections and settings and the CAN Bus Monitor (PC based software) settings. This software allows a user to interface with the Translator x2™.

## 5. Installation Instructions

---

### 5.1 Parts Shipped

You should receive:

- 1 Translator x2
- USB cable
- Software CD containing the CAN Bus Monitor software and Translator x2™ USB drivers.

The Software CD contains the CAN Bus Monitor software and USB drivers for the translator.

### 5.2 Software Installation

Before you plug the Translator x2™ into the USB port on your computer for the first time, it is recommended that you install the CAN Bus Monitor software and Translator x2™ USB drivers.

- Insert the Translator x2™ software CD into your computer. Your computer should automatically recognize that there is software to install and initiate a window pertaining to that fact. If this does not occur than you will need to manually select the setup.exe file located in the root directory of the Translator x2™ software CD.
- Tell your computer to install the software and follow the prompts on the screen. Write down where you are installing the software. This is necessary for driver Installation.
- After Installation is complete plug the Translator x2™ into the USB port on your computer. The computer will automatically try to find the necessary drivers and fail.
- After failure your computer will ask you to point to the locations of the drivers.
- Point the computer to:

..\the location where you earlier install the software\inf

If you installed the software in its default location then point to:

C:\Program Files\DakotaSoft\CAN Bus Monitor\inf

- This will install the proper drivers for your computer in the proper location.
- When complete, you are ready to start the CAN Bus Monitor software and start using the Translator x2™.

## 6. Operational Overview

---

The Translator x2™ is a CAN and Serial diagnostic device that is equipped with two CAN ports, 1 serial port and 2 USB ports. Its USB device port is used to relay CAN messages to the CAN Bus Monitor program while also allowing the user to configure the Translator's setting. The USB host port allows the user to transfer log files from the Translator's uSD Card to a thumb drive. The logged files on the uSD Card can also be accessed through Windows Explorer via the USB device port. The files on the uSD Card are saved under a FAT32 file system.

The CAN Bus Monitor software displays the raw data that is received on each CAN port and the serial port in real time. The time stamp displayed with each message is supplied by the Translator x2™. The Translator x2™ time stamp clock runs at 1ms and thus the time stamps for transmitted and received messages are extremely accurate and have none of the latency problems associated with Windows applications.

CAN messages and serial messages can be logged to a .txt file on the PC during operation or to a uSD Card on the in the Translator x2™. CAN message filters can be setup on the Bus Monitor and uploaded to the Translator x2™. Since the filters are saved to flash memory, they will be active at power up even if the Bus Monitor program is not running. The filters allow for a subset of the potential message that are on the CAN bus to be received by the Bus Monitor program.

The Translator x2™ can be setup to act as a message gateway between 2 dissimilar protocols and/or CAN networks. Parameters received in messages on 1 CAN bus can be extracted and re-encoded for broadcast on the other CAN bus. This feature will operate even when the PC is not attached to the Translator x2™.

## 7. Enclosure

---

- LEDs on the front of the enclosure give indication of when the Translator x2™ is powered, the USB host and device ports are active, there is serial port traffic and CAN port traffic.
- DB9 – 1 (CAN 1), optional 6 – 38 VDC power, 1 digital input
- DB9 – 2 (CAN 2), optional 6 – 38 VDC power, 1 digital input
- DB9 – 3 (Serial), full RS232 input
- USB female B – Full speed USB device port
- USB female A – Full speed USB host port
- Material – Glass filled, high temperature polymer
- Dimensions: 4.5 x 2.5 x 1.187 inches, 114.3 x 63.5 x 30.16 mm
- The cavity of the enclosure can optionally be filled with a weatherproofing epoxy (potting compound)

## 8. Hardware Specifications

---

### 8.1 Connectors

The Translator x2™ contains 3 male DB9 connectors, 1 series B USB device connector and 1 USB female A connector.

#### 8.1.1 CAN Port 1– DB9 Pinout





**CAN port 1 connector pinout**

Pin 1 – Power 5V	Pin 2 – CAN Low (-)	Pin 3 – NC
Pin 4 – Digital Input	Pin 5 – CAN Shield	Pin 6 – Power GND
Pin 7 – CAN High (+)	Pin 8 – NC	Pin 9 – Power (12V)

**8.1.2 CAN Port 2 & Power – DB9 Pinout****CAN port 2 connector pinout**

Pin 1 – Power 5V	Pin 2 – CAN Low (-)	Pin 3 – CAN GND
Pin 4 – Digital Input	Pin 5 – CAN Shield	Pin 6 – Power GND
Pin 7 – CAN High (+)	Pin 8 – NC	Pin 9 – Power (12V)

Note: Make sure that a 120 ohm terminating resistor is on both ends of the CAN bus. One method to determine if the bus is terminated correctly would be to measure resistance across the CAN high and CAN low wires. Your meter should read about 60 ohms (120 ohms in parallel with 120 ohms). The Translator circuitry has a built in terminating resistor on each bus. These resistors can be removed from the circuit programmatically under the “Configure Bus/CAN Port Setup” menu. See section on “CAN Ports”, for more detail.

**8.1.3 RS-232 – DB9 Pinout****CAN port 2 connector pinout**

Pin 1 – NC	Pin 2 – RS-232 Rx	Pin 3 – RS-232 Tx
Pin 4 – NC	Pin 5 – GND	Pin 6 – NC
Pin 7 – RS-232 RTS	Pin 8 – RS-232 CTS	Pin 9 – NC

**8.1.4 USB Port – Female B**

This USB port is the link between the Translator x2™ and the Bus Monitor program. Power can optionally be supplied to the Translator x2™ over the USB as well.



### 8.1.5 USB Port – Female A

This USB port is dedicated to transferring log files from the Translator's uSD card and a thumb drive.



## 8.2 Power

The Translator x2™ can be powered either from the USB port or through either of the CAN port DB9 connectors.

- DB9 provided power: 6 – 38 VDC
- USB port: 5VDC

## 8.3 Processor/Memory

- ARM Cortex (STM32F405)
- 1M flash
- 192k RAM

## 8.4 Environmental

- Operating Temperature: -40°C to +85°C
- Voltage: 6 – 38V

# 9. Communications Ports

---

## 9.1 CAN Ports

Each CAN port has a 120 ohm terminating resistor on the circuit board. Each terminating resistor can be removed from the bus circuit programmatically from the CAN Bus Monitor software, accessed through the “Configure Bus/CAN Port Settings” menu.

The baud rates for each CAN port can be changed via the CAN Bus Monitor software through the “Configure Bus/CAN Port Settings” menu.

## 9.2 1 Full Speed USB Device Port

The USB device port is the link between the Translator x2™ and the Bus Monitor program. CAN bus messages are conveyed to the Bus Monitor while the Bus Monitor uses it to configure and upload new application software to the Translator x2™

## 9.3 1 Full Speed USB Host Port

The USB host port is used to transfer log files from the Translator’s uSD Card to an external thumb drive.

## 9.4 1 RS232 Port

This port used to monitor and log traffic on the RS232 bus. The port’s baud rate can be adjusted from the “RS-232” tab.

# 10. USB Host Port

---

The USB host port allows the user to transfer log files stored on the Translator’s uSD Card to a thumb drive. Inserting a thumb drive into the host port will cause the Translator x2™ to automatically build the following directory tree:

“Root Drive Letter” (ex. G:) \TS2\Logs\ “incremental folder number (ex. 0001)” \ “log file” (ex. 00100000.txt)

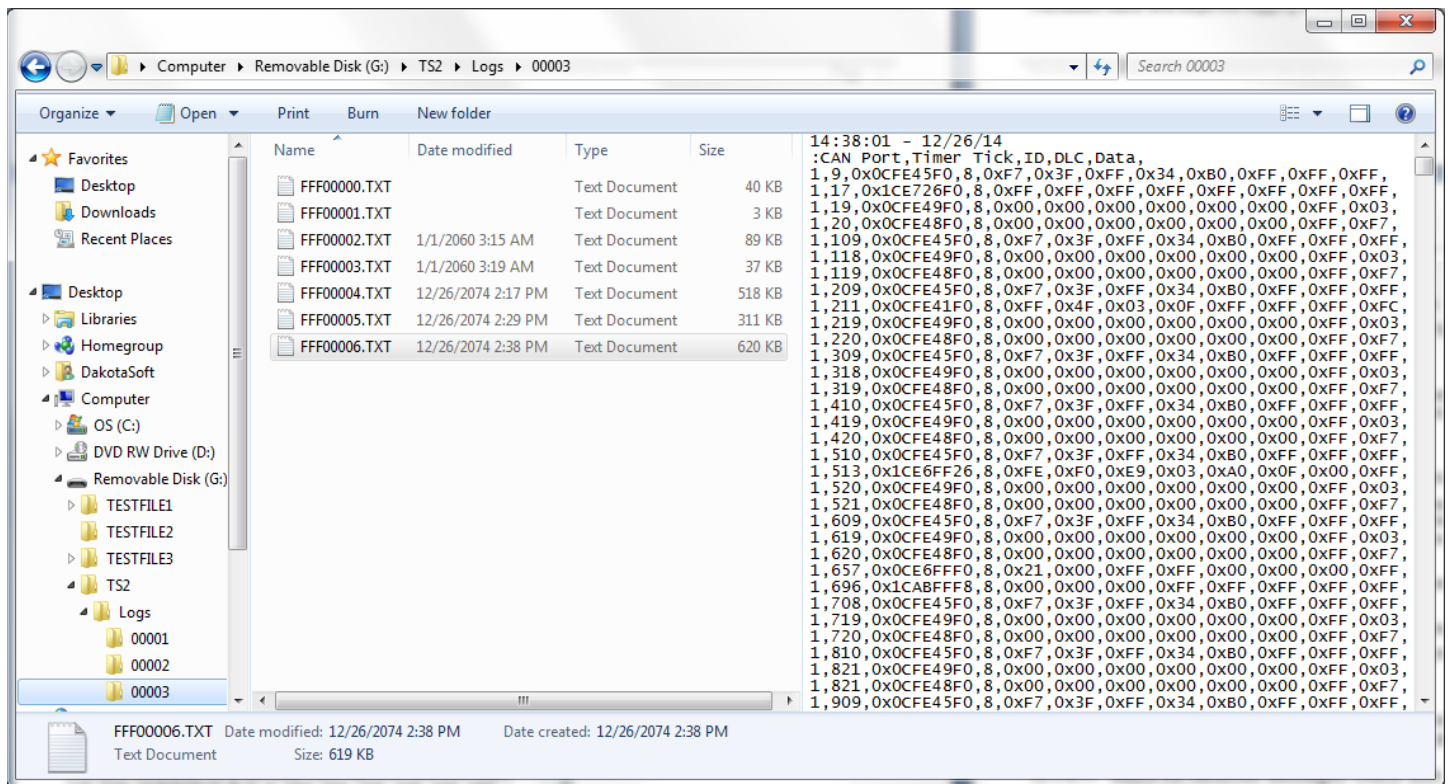


Figure 10 – Thumb drive directory structure

Files will be transferred from the Translator's uSD Card one at a time and erased from the uSD Card once it has finished transferring to the thumb drive. No data will be lost if the thumb drive is removed during a transfer.

## 11. USB Device Port

The USB device port operates as the link between the Translator x2™ and the CAN Bus Monitor software running on a PC.

## 12. CAN Bus Monitor

The CAN Bus Monitor is a PC based program that, via the USB device port, receives/transmits raw CAN messages from/to the Translator x2™. The Bus Monitor also allows the user to configure the Translator x2™, upload new application software and access the µSD Card.

### 12.1 Main Window

The CAN Bus Monitor window (Fig 12.1) contains 5 tabs that allow for the various operations of the program.

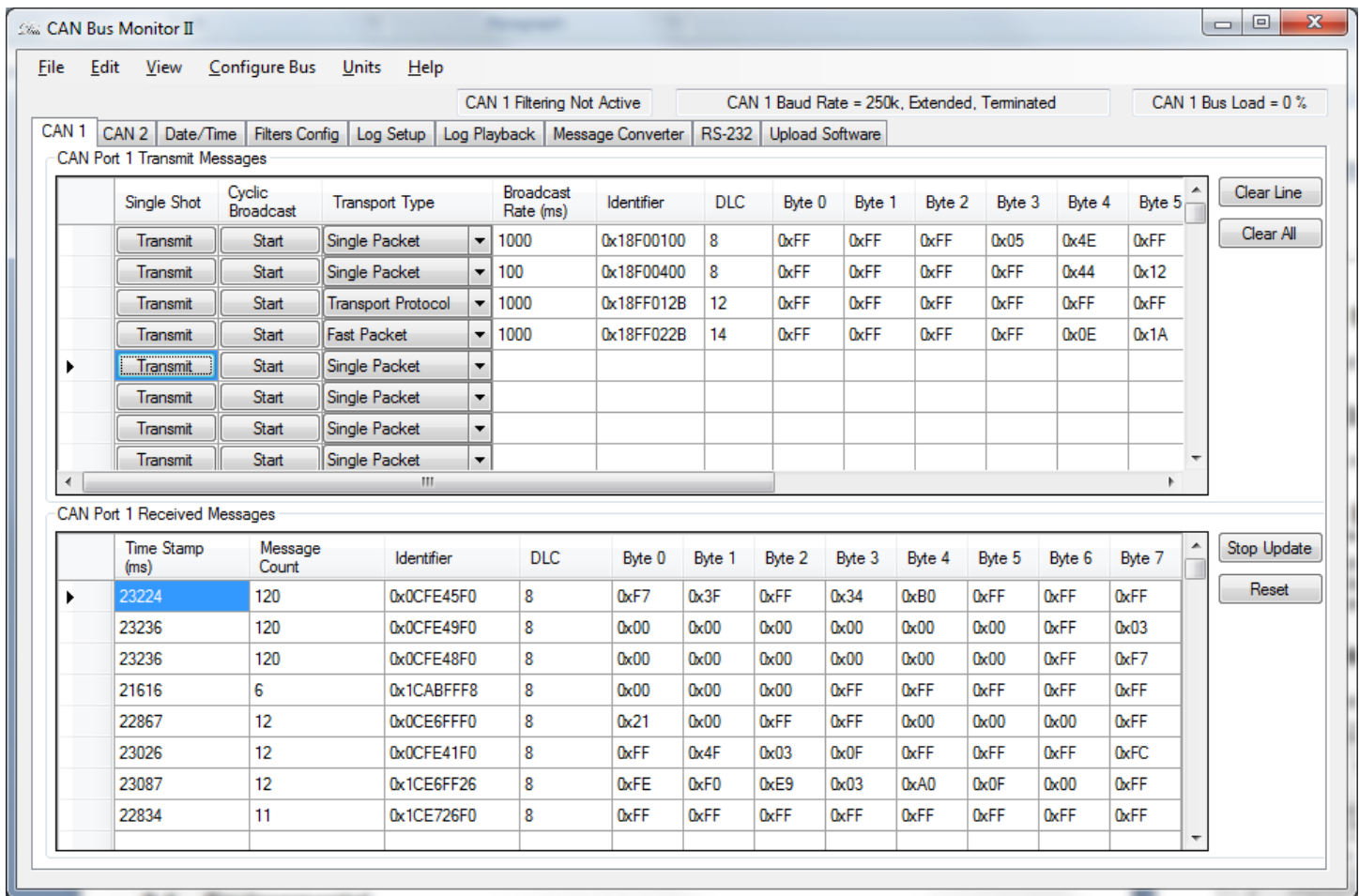


Figure 12.1 – CAN Bus Monitor CAN 1 port tab

## 12.1.1 CAN 1 & CAN 2 Port Tabs

These tabs contain the received and transmitted message data for CAN ports 1 & 2. Transmitted messages can be built in the CAN Port # Transmitted Messages window.

Transmitted messages are routed back to the Translator's receive ports. This means that any message transmitted by the Translator x2™ is also received by the Translator x2™ and displayed in the received messages window on the Bus Monitor.

### 12.1.1.1 Clear Line Button

The **Clear Line** button (Fig 12.1) allows the user to remove the currently selected CAN message from the transmit data grid. This will in turn stops the transmission of the message on the CAN bus.

### 12.1.1.2 Clear All Button

The **Clear All** button (Fig. 12.1) allows the user to remove all of the messages in the transmit data grid. This in turn stops all messages that are being transmitted by the Translator x2™ that correspond to that data grid and CAN port.

### 12.1.1.3 Transmit Button

This button allows the user to force a single transmission of a message.

#### 12.1.1.4 Cyclic Broadcast Column Start/Stop Button

The **Start/Stop** button allows the user to start/stop the cyclic broadcast of a message at a rate assigned to the message, from the **Broadcast Rate** column.

#### 12.1.1.5 Transport Type Combo Box

This box allows for setting up each message to broadcast as either a single packet, ISO multipacket or NMEA fastpacket message.

Single Packet – This message type contains a DLC anywhere from 0 to 8. Thus the data byte field will contain the number of bytes indicated by the DLC.

ISO Multipacket – This message type uses the ISO multipacket protocol to transmit the bytes in the data field. Generally speaking the DLC for this type should be greater than 8. Currently the DLC for this message type is capped at 650 bytes.

NMEA Fastpacket – This message type uses the NMEA 2000 fastpacket protocol. The DLC for this message type can range from 0 – 255 bytes.

#### 12.1.1.6 Stop Update Button

The **Stop Update** button (Fig 12.1) allows the user the freeze the update of the receive data grid. When the **Stop Update** button is pressed the wording changes from “Stop Update” to “Start Update”. CAN messages will still be received by the PC but will not be displayed in the data grid. Pressing the **Stop Update** button again will cause the wording on the button to change back to “Stop Update” and the data grid to continue updating as messages are received. If the Bus Monitor software is actively recording the receive messages, pressing the **Stop Update** button does not stop the recording.

#### 12.1.1.7 Reset Button

The **Reset** button (Fig 12.1) allows the user to clear the data grid of all the messages it has received to date. It will also reset the received message count.

#### 12.1.1.8 CAN Filter Active Window

The CAN filter active window gives the user indication of the status of the CAN message filtering on the Translator x2™. Fig. 10.1.1 shows that the Translator x2™ currently does not have its filters active. As you switch between the CAN 1 and 2 port tabs the information displayed in this window will change to display the information that correlates to the currently active port tab.

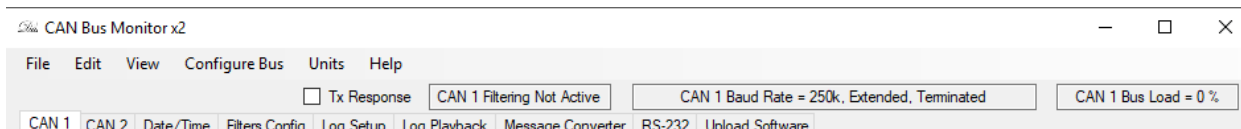


Figure 12.1.1 – CAN Bus Monitor CAN port status windows

#### 12.1.1.9 Tx Response Check Box

When checked the reception of an ISO Request Message will cause the software to generate the requested message if that message is loaded into the Tx Messages portion of the CAN Port tab.

#### 12.1.1.10 Current Baud Rate Window

This window gives indication of the baud rate for the CAN port tab that is currently active. Fig. 12.1.1 shows that CAN 1 port is currently set to run at 250k baud, extended bus type (29-bit) and Translator’s internal bus terminator is active. As you switch between the CAN 1 and 2 port tabs the information displayed in this window will change to display the information that correlates to the currently active port tab.

#### **12.1.1.11 Bus Load Window**

This window give indication of the bus load for the CAN port tab that is currently active. Fig. 12.1.1 shows that CAN 1 port is currently running with a bus load of 0%. As you switch between the CAN 1 and 2 port tabs the information displayed in this window will change to display the information that correlates to the currently active port tab.

### 12.1.2 Date/Time Tab

This tab allows the user to set the time and date of the real time clock chip on the Translator. This clock is powered by a coin cell battery that has a projected life of 10 years. The battery can be replaced by removing the top cover from the Translator x2™ and extracting the batter from its holder.

Figure 12.1.2 shows the contents of this tab. The calendar farthest to the left in the tab is the Translator's current clock setting. The calendar in the center is the PC's current clock setting. The calendar to the far right of the tab is means by which the clock can be set to a setting other than that of the PC.

Pressing the **Synchronize Translator With PC** button will cause the Translator's Date/Time settings to match that of the PC that is connected to the Translator.

The user can also manually enter a Date and Time from the **Manually Set Translator Date/Time** calendar. Pressing the **Manually Set Translator** button will cause the time set in the manual calendar to be downloaded to the Translator x2™.

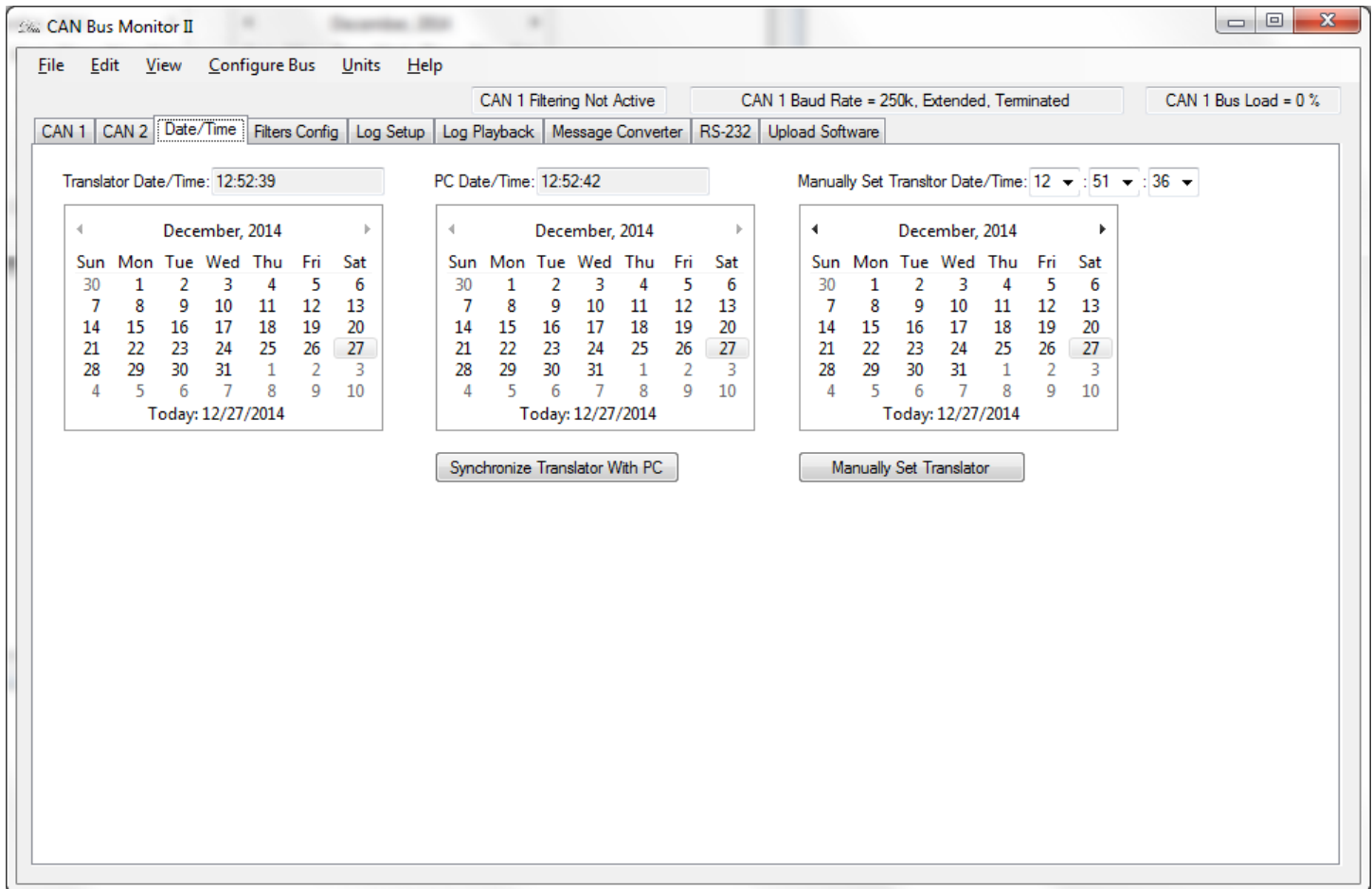


Figure 12.1.2 – Can Bus Monitor Date/Time tab



### 12.1.3 Filter Config Tab

This tab allows the user to develop and apply filters to each of the CAN ports and turn setup gateway functionality.

The filters are uploaded to the Translator x2™ where, the messages received on the CAN buses are filtered before they are delivered to the CAN Bus Monitor program for display. Fig. 12.1.2 shows the Filters tab with an active filter.

The Gateway allows for messages received on one CAN port to be rebroadcast on the other.

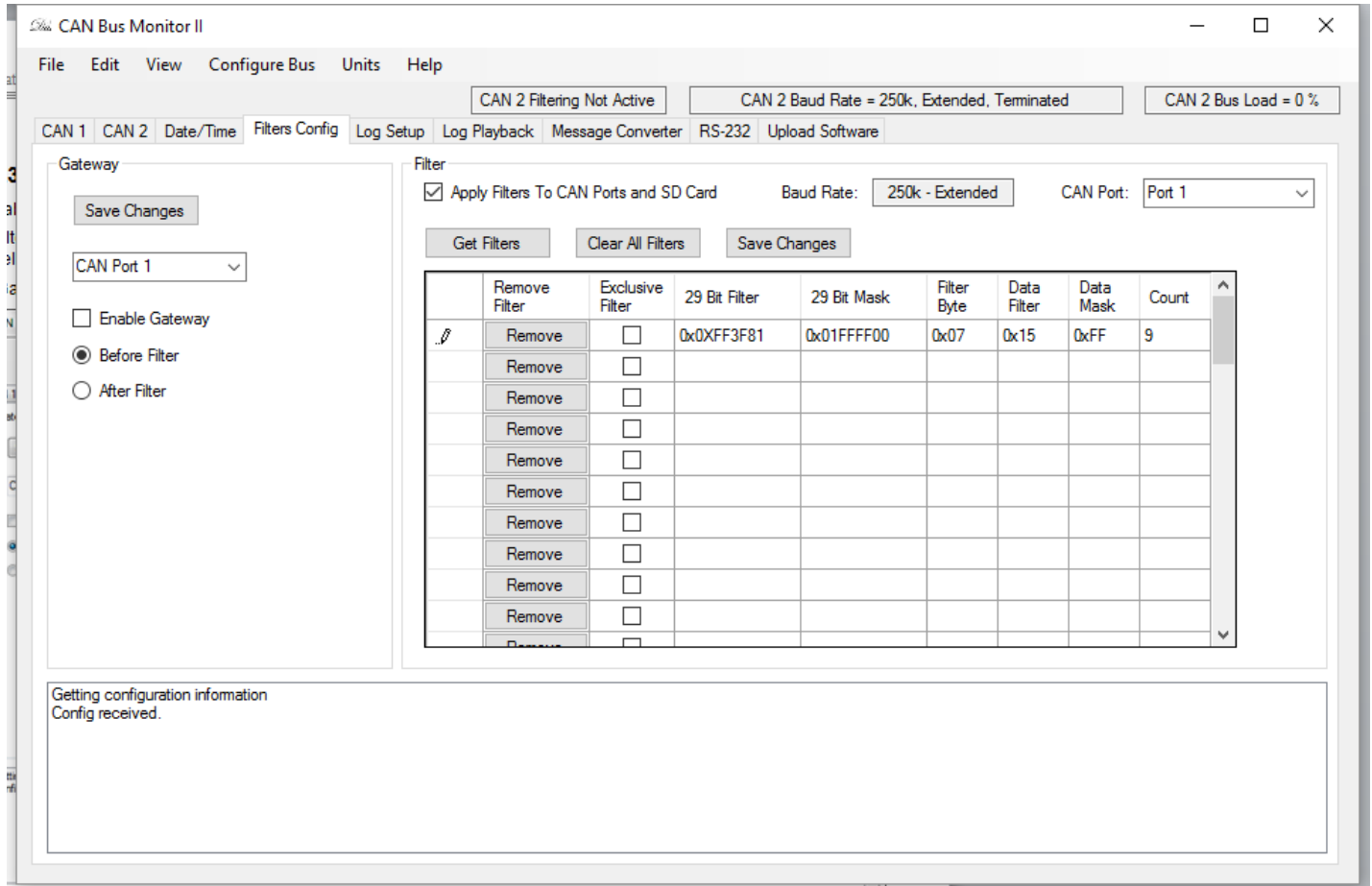


Figure 12.1.3 – CAN Bus Monitor filters config tab

#### 12.1.3.1 Filter Theory Of Operation

Pressing the Get Filters button will cause the Bus Monitor software to retrieve the CAN message filter data stored in the Translator x2™. These filters can either be modified or new ones can be created from scratch. In the Filters data grid there are 7 columns. The 1<sup>st</sup> column (Remove Filters) contains buttons that when pressed will remove the filter data entries to the right of the button.

The 2<sup>nd</sup> column (Exclusive Filter) contains a check box that allows for the setting of the filter as either inclusive or exclusive. An inclusive filter allows any messages that match the filter to pass through. An exclusive filter causes any message that match the filter to not pass through.

The 3<sup>rd</sup> column (29 Bit Filter) contains the CAN message IDs that you wish to receive. The 4<sup>th</sup> column (29 Bit Mask) contains the masks that are applied to the message IDs in the second column. Any bits set to 1 in the mask column will require any message that is received to contain the exact bit setting as the message ID in the 3<sup>rd</sup> column.

The 5<sup>th</sup> column (Filter Byte) allow for a filter to be set on one of the data bytes in the body of the message. This column sets which data byte is to be filtered. The 6<sup>th</sup> and 7<sup>th</sup> columns (Data Filter and Data Mask) operate just as the 3<sup>rd</sup> and 4<sup>th</sup> column only on the data field.

The 8<sup>th</sup> column (Count) allows the user to choose how many messages will be skipped before one is allowed to be displayed on the CAN received tab and saved to the log file. As setting of 9 will save every 10<sup>th</sup> messages that passes through the filter.

In the example in Fig. 12.1.3, only message IDs with the 2<sup>nd</sup> and 3<sup>rd</sup> bytes equaling 0xFF3F and the 7<sup>th</sup> data field equaling 0x15 will be allowed through the filters. Then the count column limits the data that is displayed or saved to 1 of every 10 received.

You must select the Apply Filters To CAN Ports and SD Card check box for the filters to become active on the Translator x2™. You can setup both CAN port's filters by using the CAN Port combo box at the top right of the Filters window to change between the filter settings for each CAN port. After the filter settings have been changed on the Bus Monitor you must then send those settings to the Translator x2™. This is done by pressing the Save Changes button. The Status window below the Filters window shows the status of the new filter upload to the Translator x2™.

µSD Card logging must be activated in order for logging of data to the µSD Card to occur. This is achieved by selecting the Activate SD Card Logging check box and pressing the Save Changes button from within the SD Card group box.

#### **12.1.3.2 Get Filters Button**

This button queries the Translator x2™ for the filters that are currently stored in its memory. These filters are automatically retrieved from the Translator x2™ when the Bus Monitor program is opened or the Translator x2™ is attached to the PC's USB port. However this button gives the user another means of retrieving what is in memory. Especially if they wish to verify that the changes they have made to the filters is actually what is in memory.

#### **12.1.3.3 Clear All Filters Button**

This button clears all of the filter data contained in the filter data grid. This button does not clear the filters saved in Translator x2™ memory. This can only be done by pressing the Save Changes button after you have cleared the filters from the data grid.

#### **12.1.3.4 Save Changes Button (Filters Group Box)**

This button saves the filters that are in the filter data grid to Translator x2™ memory.

#### **12.1.3.5 CAN Port Combo Box**

This drop down menu allows the user to change the CAN port filter data that is currently viewed in the Filters group box. Each CAN port has its own filter data.

#### **12.1.3.6 Apply Filters To CAN Port and SD Card Check Box**

This check box allows the user to activate the filters that are currently stored on the Translator x2™ for each CAN port.

#### **12.1.3.7 Current Baud Rate Window**

This window gives indication of what the current baud rate setting is for the CAN port whose filter data is being displayed.

#### **12.1.3.8 Save Changes Button (SD Card Group Box)**

This button saves the µSD Card configurations to Translator x2™ memory.

#### **12.1.3.9 Gateway Theory Of Operation**

The Gateway feature allows the Translator x2™ to be setup as a junction between two CAN buses or a buffer between a device and a CAN bus. The Gateway feature allows for messages received on one CAN port to be broadcast on the other. This can be done either before and after filtering has been applied.

### 12.1.3.10 Save Changes Button (Gateway Group Box)

This button saves the Gateway configurations to Translator x2™ memory.

### 12.1.3.11 CAN Port Combo Box (Gateway Group Box)

This combo box allows the user to change between the CAN port 1 and CAN port 2 gateway setting.

### 12.1.3.12 Enable Gateway Check Box

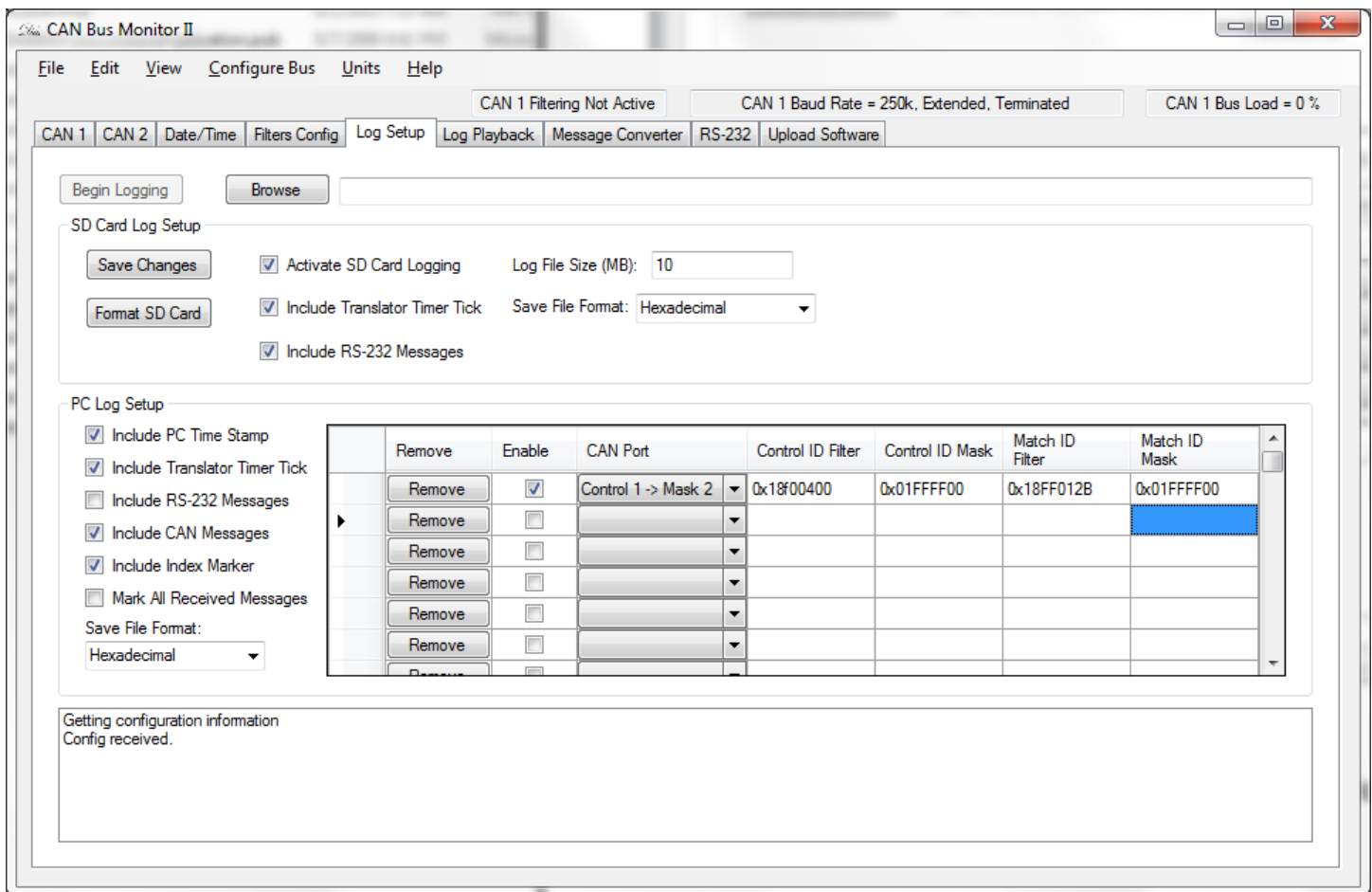
This box enables and disables the Gateway feature on the Translator x2™. The user must update the Translator x2™ with the new settings by pressing the Save Changes Button before they will take effect.

### 12.1.3.13 Before/After Filter Radio Buttons

This radio buttons allow the user to define whether the received messages must go through the filters before they are rebroadcast on the other CAN bus. If the Before Filters radio button is selected than the filters will not apply to the received messages and all received message will be broadcast on the other bus. If the After Filter radio button is selected than the filters will apply to the received messages and only the messages that pass through the filters will be broadcast on the other bus.

## 12.1.4 Log Setup Tab

The tab allows the user to set the PC and µSD Card file logging options.



### 12.1.4.1 Begin Logging Button

This button starts and stops the logging of data to the log file.

### 12.1.4.2 Browse Button

This button is used to choose the name of the file where the data is to be saved.

### 12.1.4.3 Activate µSD Card Logging Check Box

This activates the logging of CAN port messages to the µSD Card. This setting is stored in Translator x2™ memory and thus is can be active even when the PC is not attached.

When logging is active messages will be cached in memory until the set file size of data is accumulated, a thumb drive is inserted into the USB host port or the device is physically powered down. When the file size reaches the set “log file size” the file is closed and a new file is started. When the thumb drive is attached the cached data is written to the current file and the file is closed. All files on the uSD Card are then transferred to the thumb drive. Files on the SD Card are erased after they have been transferred. Once all files have be transferred a new log file is opened and bus traffic logging resumes.

See Fig 12.1.4.3 for an example of what the log file format looks like. The first line in the log file is the time and date that the log file was created.

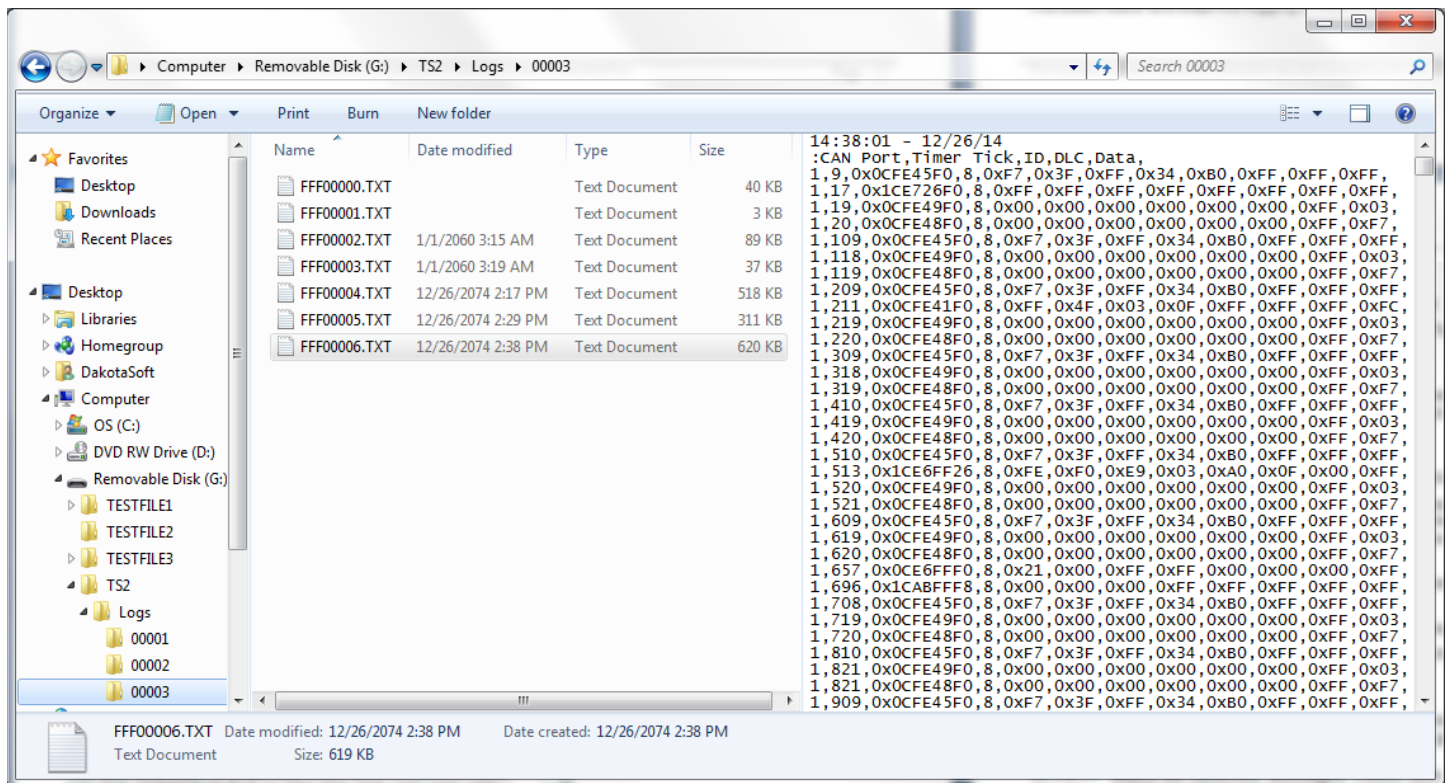


Figure 12.1.4.3 – µSD Card message log file that has been transferred to a thumb drive.

A breakdown of the first message in the above log file is as follows:

1 = CAN port 1

9 = 9ms (Translator 1ms timer tick)

0x0CFE45F0 = 29 bit ID

8 = DLC

0xF7 = Data byte 0

0x3F = Data byte 1

0xFF = Data byte 2

....

#### **12.1.4.4 Activate SD Card Logging Check Box**

This check box will cause the Translator x2™ to log bus traffic to the uSD Card. This setting is saved in the Translator's flash and will be active anytime the Translator x2™ is powered.

#### **12.1.4.5 SD Card Include Translator Timer Tick Check Box**

This check box will cause the Translator to add its timer tick to each message saved in the uSD Card log file.

#### **12.1.4.6 SD Card Include RS-232 Messages Check Box**

This check box will cause the Translator save any received RS-232 data to the log file.

#### **12.1.4.7 Log File Size (MB) Text Box**

This is where the uSD Card log file size is set. For example if the log file size is set to 20 than a new file will be started when the current log file size reaches 20MB. The default setting is 10MB.

#### **12.1.4.8 SD Card Save File Format Combo Box**

This box changes the format in which the log file data is saved. When decimal is selected all value logged will be in decimal. When hexadecimal is selected the message ID and all data bytes will be saved as hexadecimal. When National Instruments (.ncl) is selected the can messages will be saved in National Instrument's binary file format.

#### **12.1.4.9 Include PC Time Stamp Check Box**

This check box will cause the PC based log file to have a PC based time stamp.

#### **12.1.4.10 PC Include Translator Timer Tick Check Box**

This check box will cause the PC based log file to have the Translator x2™ generated timer tick.

#### **12.1.4.11 PC Include RS-232 Messages Check Box**

This check box will cause the PC to save any RS-232 messages received to the log file. Log files can have CAN and RS-232 message interlaced within the data. This is valuable if you are for example recording CAN and GPS (NMEA 0183) messages on a tractor at the same time.

#### **12.1.4.12 Include CAN Messages Check Box**

This check box will cause the PC to save any CAN messages received to the log file. Log files can have CAN and RS-232 message interlaced within the data. This is valuable if you are for example recording CAN and GPS (NMEA 0183) messages on a tractor at the same time.

#### **12.1.4.13 Index Marker Theory Of Operation**

Index markers are used to associate different messages that are received on the CAN ports.

Control filters and match filters are used to specify which message receive an index marker. When a message is received that passes through a control filter an index number is incremented and assigned to that messages. When a message is received for that control filter's corresponding match filter, the same index marker that was applied to the control filter's message is applied to the match filter's message.

The index marker is only incremented when a message passes through a control filter. Messages that don't pass through a control filter are not saved with an index marker.

Index markers are not used when the Translator x2™ is saving CAN messages to the uSD card.

Index marked log files cannot be played back using the Log Playback tab.

#### **12.1.4.14 Include Index Marker Check Box**

This check box will cause the PC based log file to have an index number recorded for every message that matches the control and match filters contained in the index marker data grid.

#### **12.1.4.15 Match All Received Messages Check Box**

This check box will cause the PC based log file to add an index marker to every message that is received. CAN port 1 is always the control CAN port and CAN port 2 is always the match CAN port. The received ID on CAN port 1 must completely match the same ID on CAN port 2. Messages with IDs on CAN port 2 that don't match any messages on CAN port 1 will not be marked with an index.

Index markers are incremented every time a message is received on CAN port 1.

If this check box is active then the index marker data grid will be inactive.

#### **12.1.4.16 Index Marker Data Grid**

This data grid allows the user to setup which messages on the CAN ports will be marked with an index number. If the mark all received messages check box is active this data grid will be inactive.

Remove Button – removes the control and mask information from the list of filters.

Enable Check Box – enables/disables the filter.

CAN Port – sets the CAN port where the control filter and match filter will be applied.

Control ID Filter – the ID filter applied to received messages on the control CAN port.

Control ID Mask – the mask applied to the control ID filter.

Match ID Filter – the ID filter applied to the received messages on the match CAN port.

Match ID Mask – the mask applied to the match ID filter.

#### **12.1.4.17 PC Save File Format Combo Box**

This box changes the format in which the log file data is saved. When decimal is selected all value logged will be in decimal. When hexadecimal is selected the message ID and all data bytes will be saved as hexadecimal. When National Instruments (.ncl) is selected the can messages will be saved in National Instrument's binary file format.

### **12.1.5 Log Playback Tab**

This tab allows the user to playback a previously recorded log file. Fig. 12.1.5 shows a file being played back.

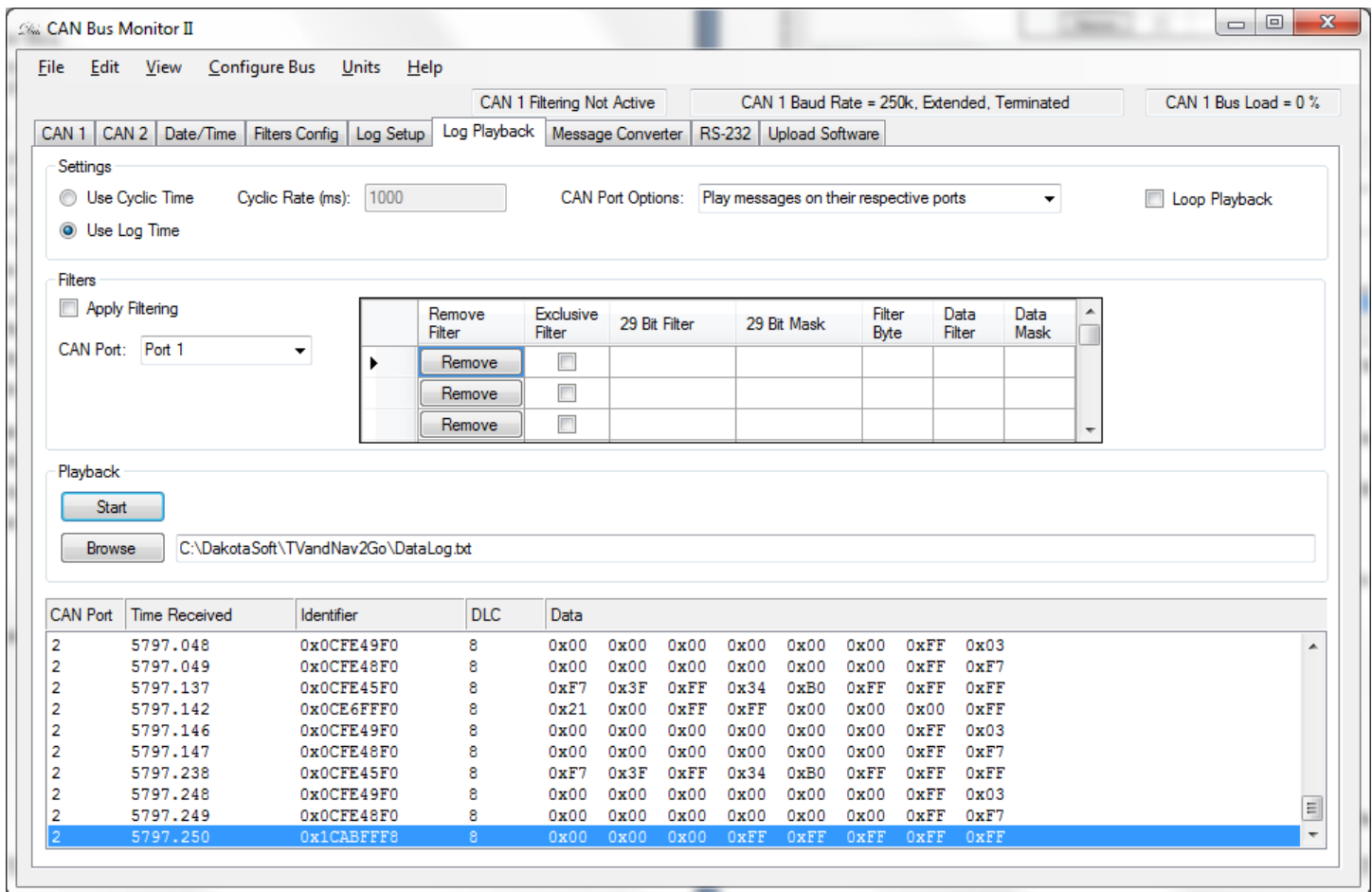


Figure 12.1.5 – Log file playback tab

### 12.1.5.1 Settings: Use Cyclic time/Use Log Time Radio buttons

The Use Cyclic Time radio button will cause the playback of the log file messages in reference to the time that the messages were received.

The use cyclic time radio button will cause the playback of the log file messages at the rate indicated in the Cyclic Rate (ms) text box. The value entered in the box is in milliseconds.

### 12.1.5.2 Settings: CAN Port Options Combo Box

This box allows for the user to changed what CAN port the logged messages are played on. The options are as follows:

- Play only port 1 messages on port 1
- Play only port 1 messages on port 2
- Play only port 2 messages on port 1
- Play only port 2 messages on port 2
- Play messages on their respective ports
- Play messages on opposite ports
- Play all messages on port 1
- Play all messages on port 2

### 12.1.5.3 Loop Playback Check Box

The loop playback check box causes the playback of the file to start over at the beginning of the file after the enter file as been played.

#### 12.1.5.4 Filters: Apply Filtering Check Box

This check box allows for the playback filters to be applied to the playback log file. When this box is checked filtering is active.

#### 12.1.5.5 Filters: CAN Port Combo Box

This combo box allows for the setting of filters for both CAN ports.

#### 12.1.5.6 Filters: Data Grid

This box allow for the setting of filters.

The Exclusive Filter check box allow for the setting of the filter as either inclusive or exclusive. An inclusive filter will let any message that matches the filter to pass through. An exclusive filter will cause any message that matches the filter to not get through.

The 29 Bit Filter field is the filter for a message or range of messages. The bits set in this field bit state that each message must contain in order to pass through the filter.

The 29 Bit Mask field contains the significant bits in the bit filter.

The Filter Byte is the byte field in the body of the message where a filter is to be applied. User 0 based counting for this value.

The Data Filter and Data Mask operate in the same fashion as the 29 Bit Filter and 29 Bit Mask.

#### 12.1.5.7 Playback: Start/Stop Button

This button causes the playback of the log file to be started and stopped.

#### 12.1.5.8 Playback: Browse Button

The browse button allows the user to select a log file for playback. To start the playback at the beginning of the log file just select the same log file for playback using the browse button.

#### 12.1.5.9 Log File Playback Window

This window displays all of the CAN bus traffic for both buses while the playback is underway.

### 12.1.6 Message Converter Tab

This tab allows the user to extract parameters from received messages, re-encode them into different messages and broadcast them on a CAN bus. For example, one might wish to extract the J1939 engine speed parameter and rebroadcast that parameter using a proprietary PGN and NMEA2000 fastpacket protocol.

Multiple parameters from different CAN messages can be concatenated into a single message or a single messages parameters can be broken up into different messages.

These conversions are saved on the Translator x2™ and are active even when a PC is not attached.

Figure 10.1.6.1 is an excerpt of the Message Conversion tab. It only shows the left half of the data grid. Figure 12.1.6.2 is an excerpt of the right half of the Message Conversion tab data grid.



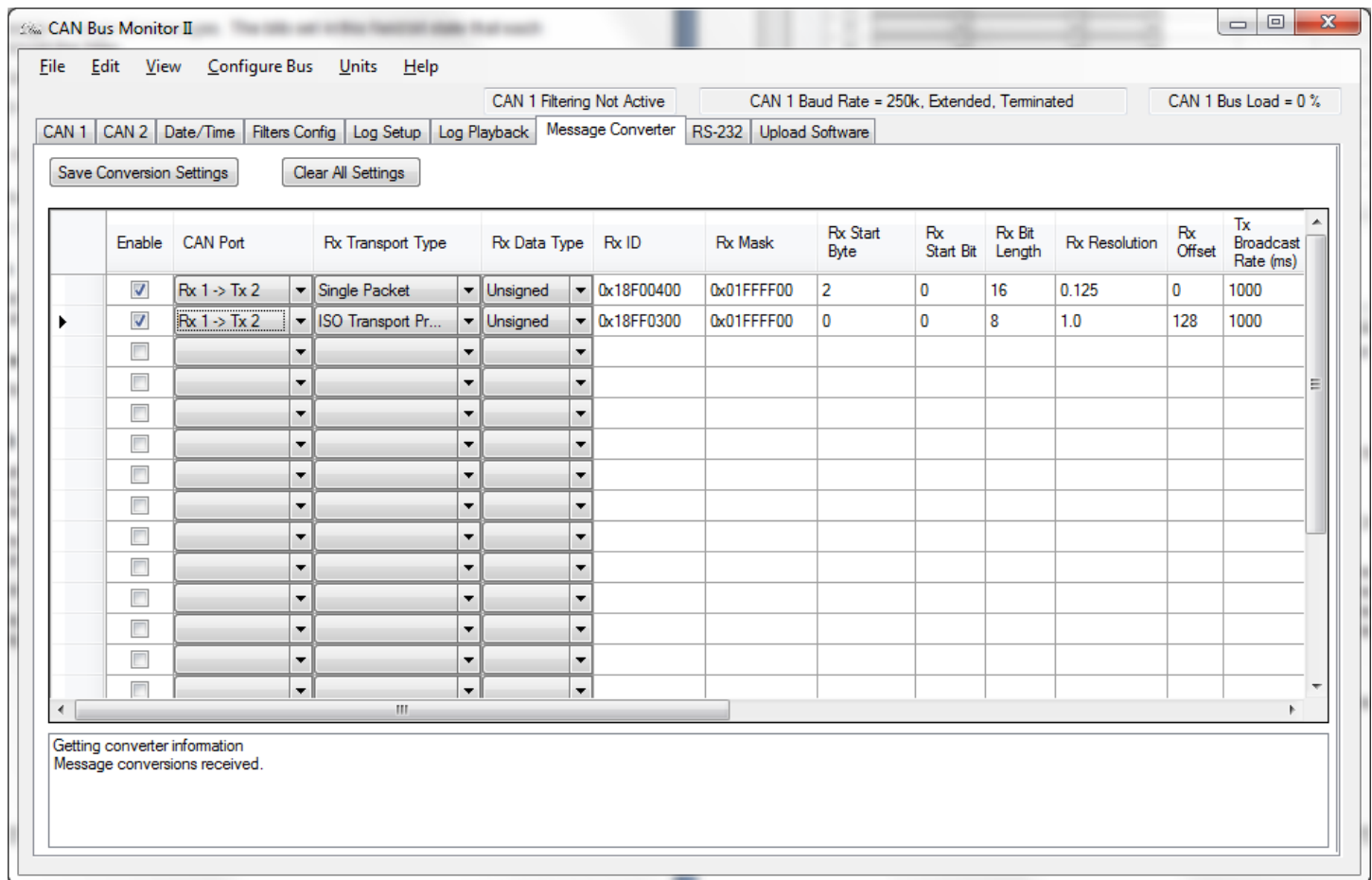


Figure 12.1.6.1 – Left side of the Message Conversion tab data grid.

### 12.1.6.1 Enable Check Box Column

This column allows for the conversion information to the right of the check box to be active.

### 12.1.6.2 CAN Port Combo Box Column

This column defines what CAN port the message is to be received on and what CAN port the resulting message is to be broadcast on. The selections are as follows:

Rx 1 -> Tx 1 – Receive the message on CAN port 1 and Transmit the resulting message on CAN port 1.

Rx 1 -> Tx 2 – Receive the message on CAN port 1 and Transmit the resulting message on CAN port 2.

Rx 2 -> Tx 1 – Receive the message on CAN port 2 and Transmit the resulting message on CAN port 1.

Rx 2 -> Tx 2 – Receive the message on CAN port 2 and Transmit the resulting message on CAN port 2.

### 12.1.6.3 Rx Transport Type Combo Box Column

This column is the type of transport protocol that is used to receive the CAN message. The options are as follows:

Single Packet – ISO single packet message.

ISO Multipacket Protocol – ISO multipacket protocol.

NMEA Fastpacket – NMEA 2000 fastpacket protocol.

#### **12.1.6.4 Rx Data Type Combo Box Column**

This column is the data type of the received parameter. The options are as follows:

Signed  
Unsigned

#### **12.1.6.5 Rx ID Column**

This column is the ID of the CAN message that is to be received.

#### **12.1.6.6 Rx Mask Column**

This column is the mask for the received message ID column. Bits set to 1 in the in this column cause the same bits in the Rx ID column to be significant.

#### **12.1.6.7 Rx Start Byte Column**

This column is the byte within the body of the message where the parameter starts. This uses 0 based counting.

#### **12.1.6.8 Rx Start Bit Column**

This column is the bit within the starting byte of the message where the parameter starts. This uses 0 based counting and cannot be greater than 7.

#### **12.1.6.9 Rx Bit Length Column**

This column is the length of the parameter in bits.

#### **12.1.6.10 Rx Resolution Column**

This column is the value that is multiplied times the received parameter. The value is in floating point.

#### **12.1.6.11 Rx Offset Column**

This column is the value that is subtracted from the parameter after the resolution is applied. This is a whole number.

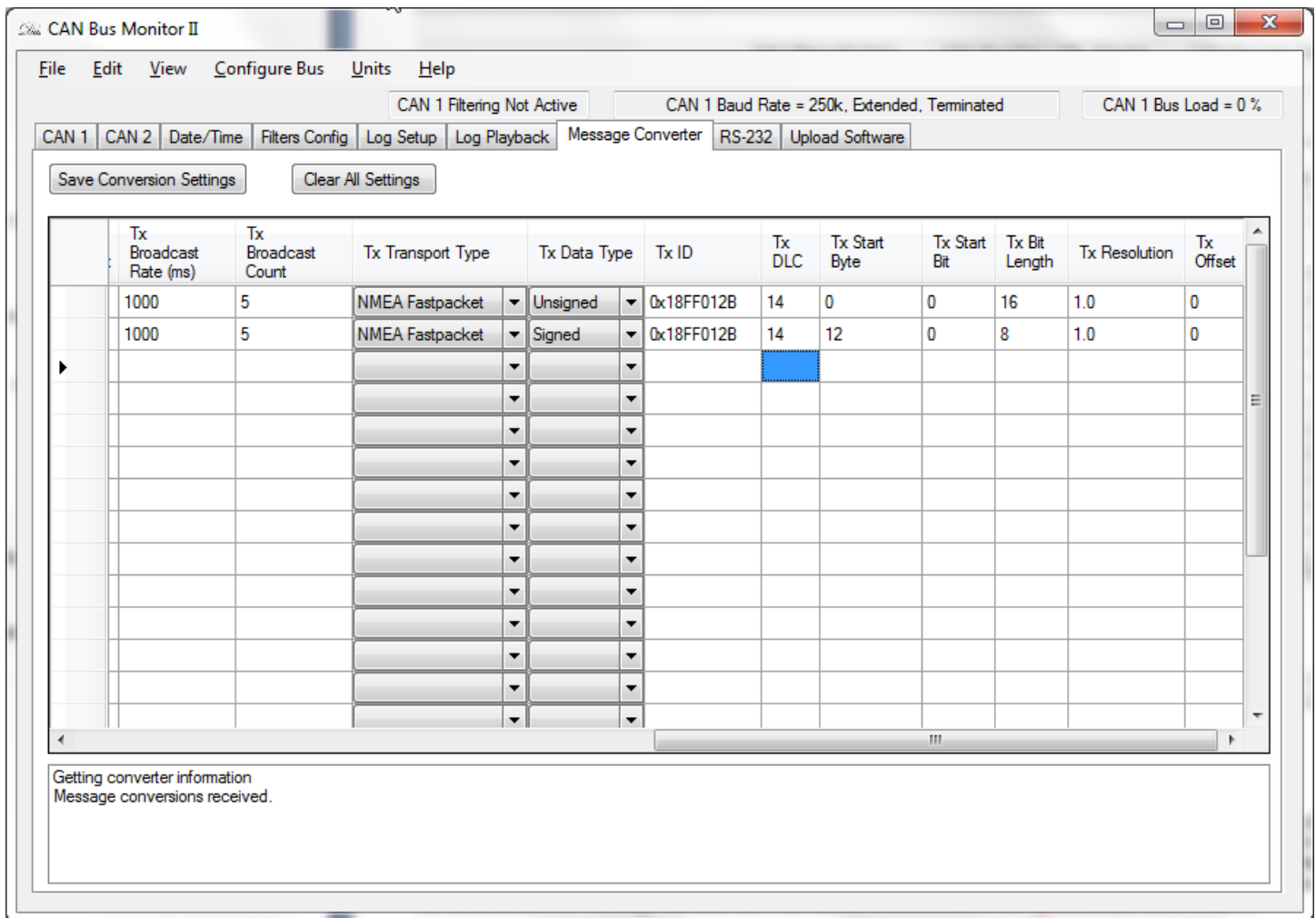


Figure 12.1.6.2 – Right side of the Message Conversion tab data grid.

#### 12.1.6.12 Tx Broadcast Rate (ms) Column

This column works in conjunction with the Tx Broadcast Count column. It is the length of time that the Translator x2™ will wait before it broadcasts the resulting CAN message. When the Translator x2™ receives the first CAN message it is looking for it will start a timer for that messages. When the timer elapses the Translator x2™ broadcasts the resulting message and resets the timer. It will again wait until the timer has elapsed before broadcasting the message again. This will continue until the message has been sent the number of times listed in the Tx Broadcast Count column.

### 12.1.6.13 Tx Broadcast Count Column

This column works in conjunction with the Tx Broadcast Rate column. It allows the resulting message to be broadcast a given number of times after the received message is captured. Every time a receive message is captured the Tx Broadcast Count gets reset to its original value.

#### 12.1.6.14 Tx Transport Type Column

This column defines how the resulting message will be transmitted. It is the same as the Rx Transport Type. Its options are as follows:

Single Packet – ISO single packet message.  
ISO Multipacket – ISO multipacket transport protocol.  
NMEA Fastpacket – NMEA 2000 fastpacket protocol.

#### **12.1.6.15 Tx Data Type Column**

This column defines how the parameter data is to be treated. It is the same as the Rx Data Type. Its options are as follows:

Signed  
Unsigned

#### **12.1.6.16 Tx ID Column**

This column is the ID of the CAN message that is to be transmitted.

#### **12.1.6.17 Tx DLC Column**

This column is the number of bytes in the body of the message. This influences how many bytes are broadcast for a given message.

#### **12.1.6.18 Tx Start Byte Column**

This column is the byte within the body of the message where the parameter starts. This uses 0 based counting.

#### **12.1.6.19 Tx Start Bit Column**

This column is the bit within the starting byte of the message where the parameter starts. This uses 0 based counting and cannot be greater than 7.

#### **12.1.6.20 Tx Bit Length Column**

This column is the length of the parameter in bits.

#### **12.1.6.21 Tx Resolution Column**

This column is the value that is multiplied times the received parameter. The value is in floating point.

#### **12.1.6.22 Tx Offset Column**

This column is the value that is subtracted from the parameter after the resolution is applied. This is a whole number.

### **12.1.7 RS-232 Tab**

This tab allow the user to setup the Translator x2™ to received data over the RS-232 port. Different baud rates and different modes of operation can be set. All setting are stored in the Translator x2™ flash memory and are thus operational when the Translator x2™ is operated without a connection to a PC.

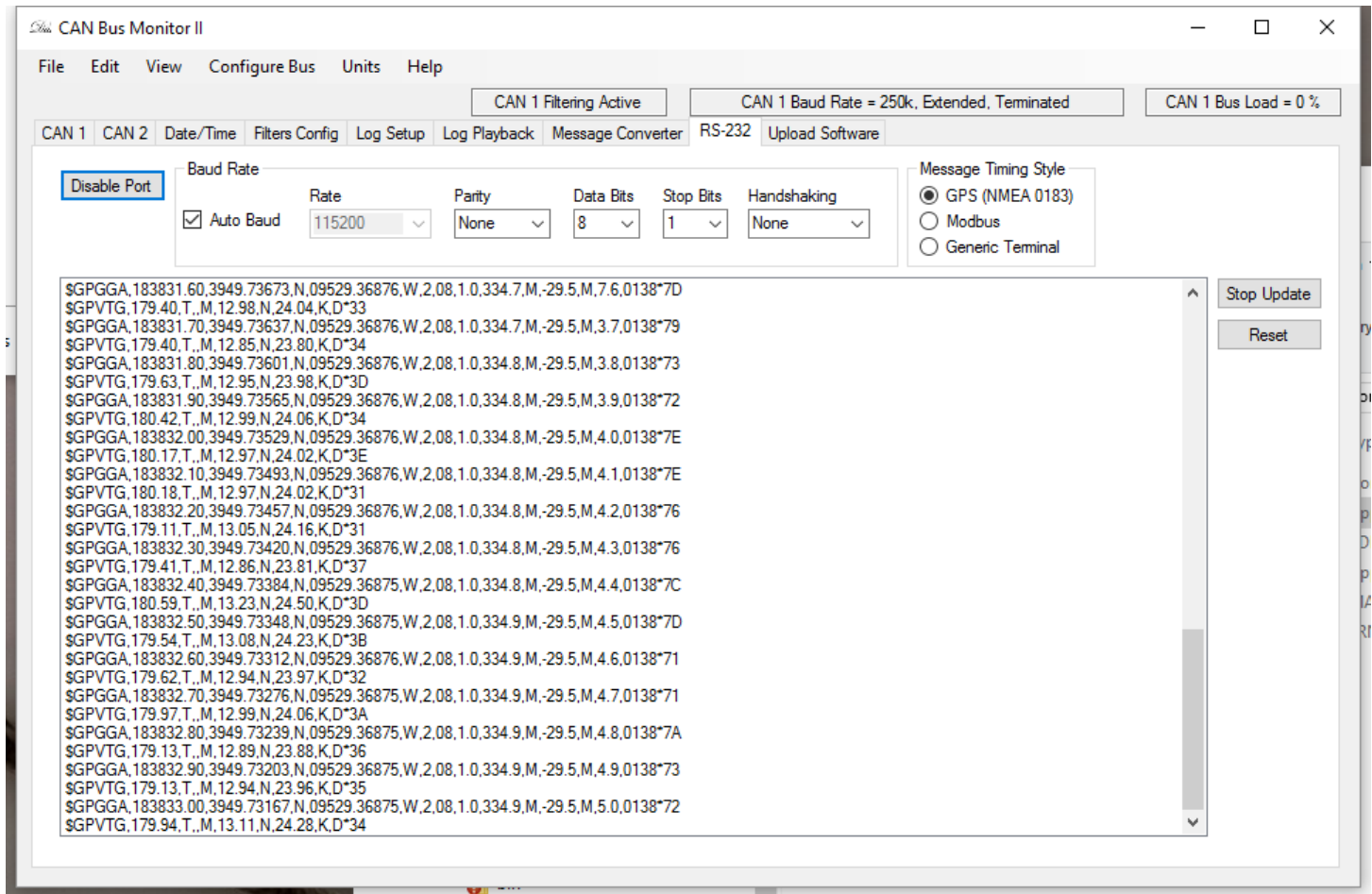


Figure 12.1.7 – CAN Bus Monitor II RS-232 tab

### 12.1.7.1 Baud Rate

The baud rate settings allow you to choose the baud rate, parity, number of data bits, number of stop bits and whether handshaking is implemented.

The allowable baud rate settings are:

2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200

The allowable parity settings are:

None, even, odd

The allowable data bit settings are:

8, 9

The allowable stop bit settings are:

0.5, 1, 1.5, 2

The allowable handshaking settings are:

None, RTS, CTS, RTS/CTS

### 12.1.7.2 Auto Baud

The auto baud feature allows the Translator x2™ to determine which baud rate the bus is operating at. If this feature is selected, at power up the Translator x2™ will wait until it sees bus traffic. It will then sample the bus for 5 seconds to

determine at what baud rate the bus is operating. A baud rate will be selected from the list of available baud rates, listed in the section “Baud Rate”, that most closely matches what was sampled. Once a baud rate has been selected that rate will appear in the RS-232 tab’s “Rate” combo box and will be used until the next power cycle.

When this feature is selected it is not possible to manually change the baud rate. The auto baud feature is only available in NMEA 0183 mode.

#### **12.1.7.3 Message Timing Mode**

The message timing mode allows the operator to choose how to terminate a received message. When a message termination is determined that message is then saved to the SD Card and/or delivered to the CAN Bus Monitor software for display.

GPS (NMEA 0183) Mode: This mode determines that a message is terminated when the carriage return and line feed characters are received.

Modbus Mode: This mode determines that a message is terminated when there is a 2.5 bit time period with no activity on the bus.

Generic Terminal Mode: This mode determines that the receipt of each byte is terminated. Thus each byte is delivered to the SD Card and/or CAN Bus Monitor program as they are received.

#### **12.1.7.4 Disable Port Button**

The Disable Port button enables/disables the RS-232 port reporting from the Translator x2™ to the CAN Bus Monitor II software. The port must be enabled to log RS-232 messages to a PC based log file. It does not affect the recording of the RS-232 port data to the SD Card.

#### **12.1.7.5 Stop Update Button**

The Stop Update button stops the update of received message on the RS-232 tab’s message window. Messages received from the Translator x2™ while the update is stopped are added to the receive list internally by the CAN Bus Monitor II software. When the button is selected to resume updates, the window will populate with the current list of received messages.

#### **12.1.7.6 Reset Button**

The Reset button clears the current list of received message from the RS-232 tab’s message window.

### **12.1.8 Upload Software Tab**

This tab allows the user to upload new application code to the Translator x2™. Fig. 12.1.8 shows a new application being loaded onto the Translator x2™.

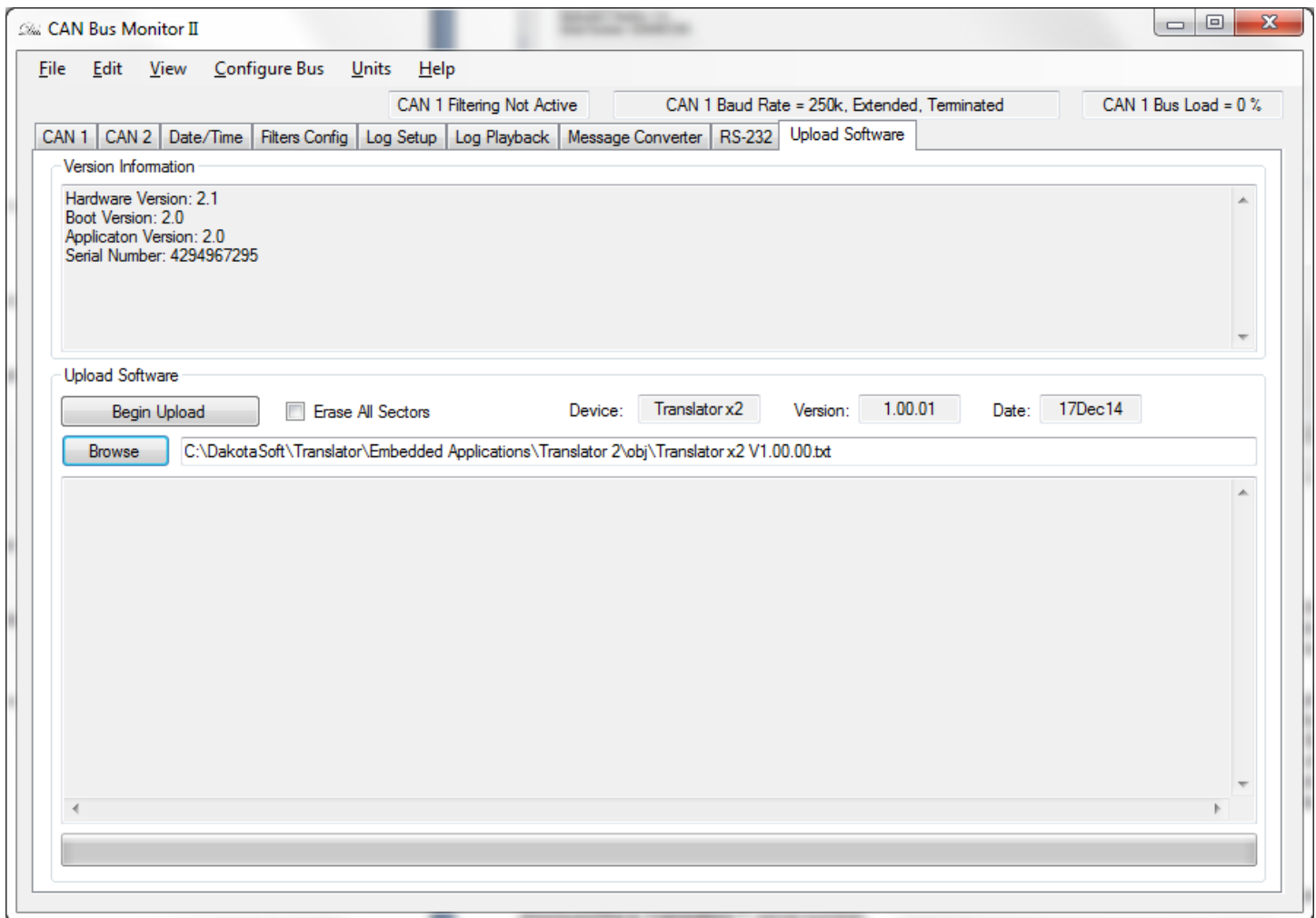


Figure 12.1.8 – CAN Bus Monitor II upload software tab

### 12.1.8.1 Theory Of Operation

Using the version information in the Version Information window the user will determine if they would like to update their application software. They can then select a .hex file containing the application code they wish to upload by using the Browse button. Once selected the path to the .hex file will appear in the window to the right of the Browse button. Once the user presses the Begin Upload button the upload application software process will begin.

The Translator x2™'s program counter will be reset to the boot block where it will remain until the application upload process is complete. If the upload process is interrupted for any reason (for example power is lost) the Translator x2™ will only operate from within the boot block until the application software is successfully loaded. The Bus Monitor program will then start to transfer the application code to the Translator x2™. The progress bar at the bottom of the screen will give indication as to how much of the upload process is left. After all of the application software is uploaded, the Bus Monitor will tell the Translator x2™ to jump to the beginning of the application software. The Translator x2™ will begin running with its new application.

### 12.1.8.2 Version Information Window

This window displays the Translators™ current hardware, boot block software and application software versions. Also displayed is the Translators™ serial number.

### 12.1.8.3 Browse Button

This button allows the user to search through their file system for the .hex file they wish to upload to the Translator x2™.

### 12.1.8.4 Erase All Sectors Check Box

Checking this box will cause all of the application code space on the Translator x2™ to be erased prior to programming. If this box is not checked, only the sectors identified in the .hex file will be erased prior to programming. Selecting this box will cause all configuration information previously upload to the Translator x2™ to be lost. The Translator x2™ will revert to its factory default configuration.

### 12.1.8.5 Begin Upload Button

This button starts the software upload process. After starting the upload process this button will be disabled until the upload process is complete or an error condition arises.

### 12.1.8.6 Status Window

The status window relays any error conditions that may arise during programming.

### 12.1.8.7 Progress Bar

The progress bar gives indication of how much time is left in the upload process.

## 12.2 File Menu

Fig. 12.2 shows the **File** menu. This menu allows the user to save configurations, load configurations and log CAN messages to a file.

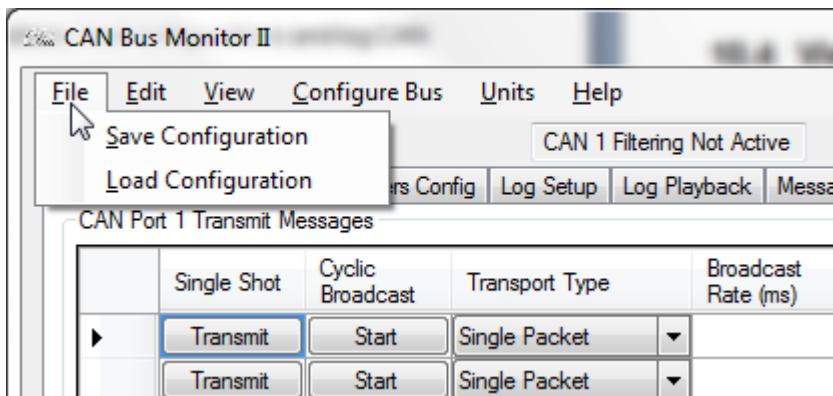


Figure 12.2 – Bus Monitor File Menu

#### 12.2.1.1 Save Configuration Menu Option

This menu options allow the user to save the current Bus Monitor configuration to a .ccfg file. Configuration data includes: Unit settings, Windows view setting and any transmitted messages entered into the Tx messages data grids.

#### 12.2.1.2 Load Configuration Menu Option

This menu option allows the user to load saved configuration data into the Bus Monitor program. Configuration data includes: Unit settings, Windows view setting and any transmitted messages entered into the Tx messages data grids.



## 12.3 Edit Menu

Fig. 12.3 shows the Edit menu. The user can select a message from the received messages window, select the Edit menu item “copy” to copy the received messages information to the clipboard. The hot keys Ctrl-C can also be used to copy the message information to the clipboard.

This information can then be pasted into the Transmit window using the menu item “Paste”. The hot key Ctrl-V can also be used to paste in information on the clipboard to the transmit window.

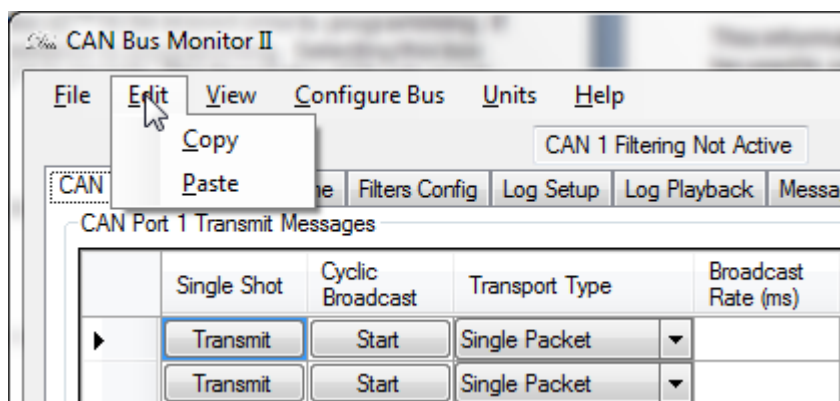


Figure 12.3 – Bus Monitor Edit Menu

## 12.4 View Menu

The CAN Bus Monitor allows the user to hide unused data grids. This is done by entering the **View** menu (Fig 12.4) and selecting the data grid that is to be hidden. The data grids with a check mark to the left of their name are currently visible. Unchecked data grids will be hidden and the visible data grids will automatically resize themselves to utilize the vacated space of a hidden data grid.

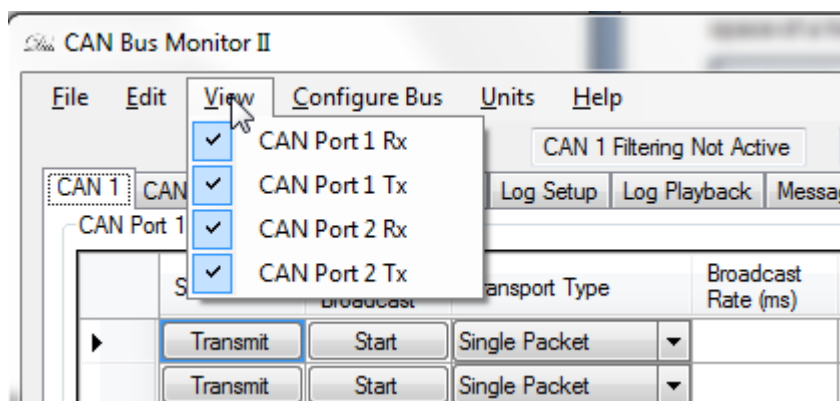


Figure 12.4 – Bus Monitor View Menu

## 12.5 Configure Bus Menu

The **Configure Bus** menu (Fig 12.5) option allows the user to change the baud rate that each CAN port is operating at, select the CAN mode (Standard or Extended) and select whether the bus is terminated or not.. Selecting this menu option will open a separate window (Fig 12.5.1) where these changes can take place.

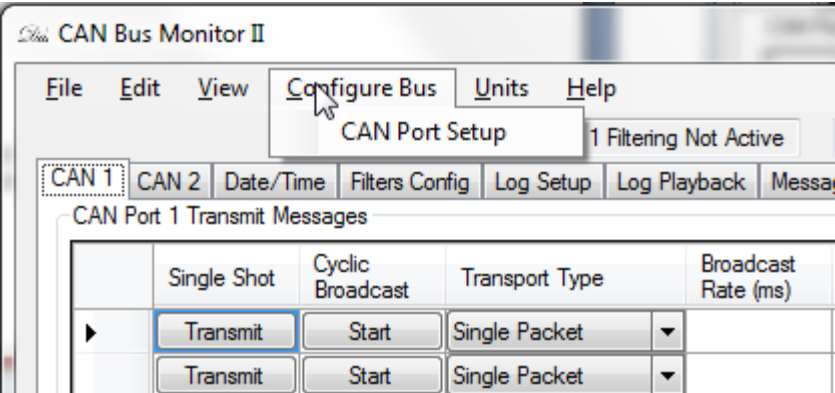


Figure 12.5 Bus Monitor Configure Bus Menu

Within the change baud rate window the user can select the CAN port whose baud rate is to be changed, select either 11 bit standard CAN or 29 bit extended CAN, select a baud rate ranging from 10kHz to 1MHz and choose to apply termination to the bus.

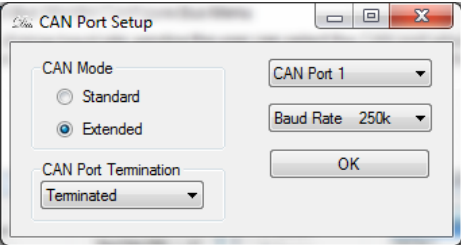


Figure 12.5.1 – Configure CAN Port

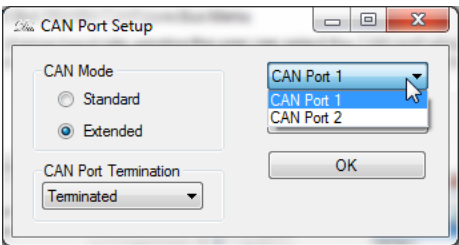


Figure 12.5.2 – Select CAN Port

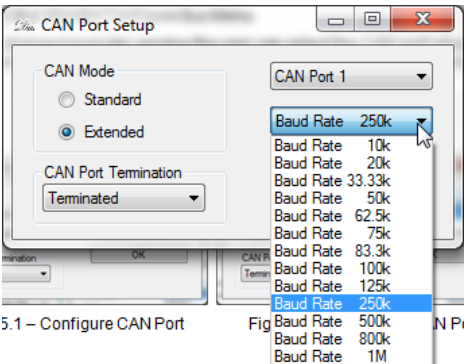


Figure 12.5.3 – Change Baud Rate

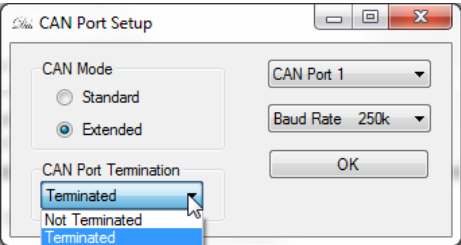


Figure 12.5.4 – Change CAN Port Termination

## 12.6 Units Menu

The **Units** menu (Fig 12.6) allows the user to change the units displayed in the data grid between decimal to hexadecimal. The check mark to the left of the menu selection indicates the current setting.

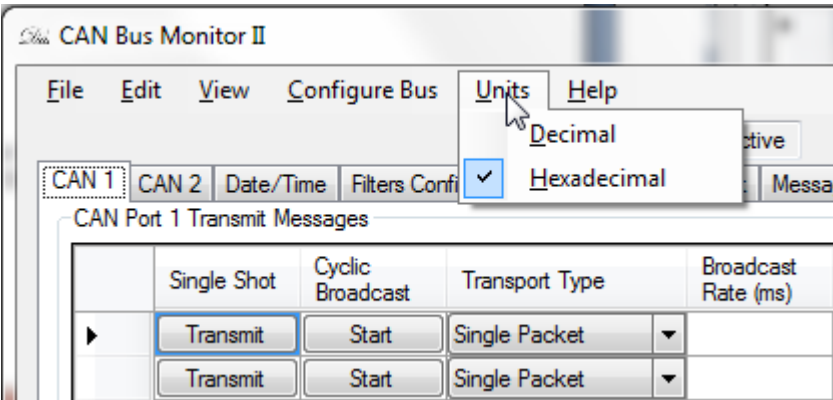


Figure 12.6 – Bus Monitor Units Menu

## 12.7 Help Menu

The help menu (Figs 12.7.0 & 12.7.1) shows the Bus Monitor's software version.

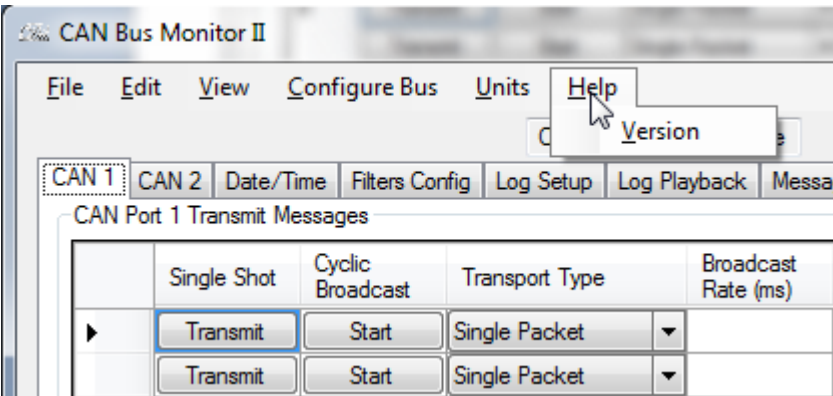


Figure 12.7.0 – Bus Monitor Help Menu

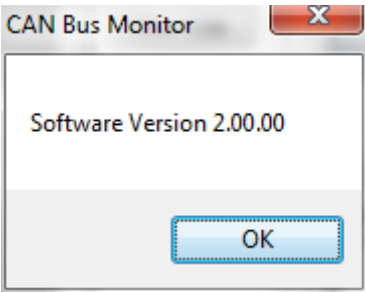


Figure 12.7.1 – Software Version



```

0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC0, 0xC1, 0x01,
0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E,
0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09, 0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F,
0xDD, 0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3, 0x11, 0xD1, 0xD0, 0x10,
0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D,
0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A, 0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA,
0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26, 0x22, 0xE2, 0xE3, 0x23,
0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4,
0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F, 0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79,
0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5, 0x77, 0xB7, 0xB6, 0x76,
0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97,
0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C, 0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98,
0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C, 0x44, 0x84, 0x85, 0x45,
0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80, 0x40,
};

```

### 13.3 Rx CAN Message

Message structure containing a received CAN message.

#### Message Structure

Byte 0: Command byte

0x80 = Rx CAN message

Byte 1: CAN port

0 = CAN port 1

1 = CAN port 2

Bytes 2 – 9: 64 bit Translator time tick of when the message was received.

Bytes 10 – 13: 29 bit ID

Bytes 14 – 15: DLC (data length code), The number of bytes in the body of the message.

Bytes 16 – n: Data bytes from the body of the CAN message.  $n = 16 + \text{DLC}$ .

### 13.4 Tx CAN Message

Causes the transmission of a message on a CAN port. The transmitted message can be single packet, multipacket or fastpacket.

#### Message Structure

Byte 0: Command byte

0x01 = Tx CAN message

Byte 1: CAN port

0 = CAN port 1

1 = CAN port 2

Bytes 2 – 3: DLC (data length code), The number of bytes in the body of the message

Bytes 4 – 7: CAN Id (11 bit ID for standard CAN, 29 bit ID for extended CAN)

Bytes 8 – 11: Delay Time, This is the time between transmitted messages that the Translator will wait. This value is in milliseconds.

For example: If the time delay for this message is 10ms, then the Translator will wait 10ms after the previous transmit message, that was loaded into the Translator, has broadcast before broadcasting this message.

Byte 12: Transport type

0 = Single packet (ISO single packet)

1 = Multipacket (ISO transport protocol)

2 = Fastpacket (NMEA fastpacket)

**Multipacket:**

Byte 13: Packet delay; This is the delay time between transmitted packets (ISO standard states a minimum of 50ms)  
n = milliseconds

Bytes 14 – n: Data bytes; ISO multipacket messages cannot exceed 50 bytes in length

**Fastpacket:**

Bytes 13 – n: Data bytes; NMEA fastpacket messages cannot exceed 223 bytes in length

**Single packet:**

Bytes 13 – n: Data bytes; ISO single packet messages cannot exceed 8 bytes in length

Response Message Structure

None

## 13.5 Reset USB Port Message

Causes the Translator to run its USB initialization functions. This will cause the Translator to perform a soft disconnect/connect to the PC.

Command Message Structure

Byte 0: Command byte

0x02 = Reset USB message

Response Message Structure

Byte 0: Command byte

0x82= Reset USB port message

Byte 1: Success/Failure

0 = Failure

1 = Success

## 13.6 Reset CAN Ports Message

Causes the Translator to run its CAN initialization functions.

Command Message Structure

Byte 0: Command byte

0x03 = Reset CAN ports message

Response Message Structure

Byte 0: Command byte

0x83 = Reset CAN port message

Byte 1: Success/Failure

0 = Failure

1 = Success

## 13.7 Set Hardware Filters Message

Command Message Structure

Byte 0: Command byte

0x04 = Set hardware filters

Response Message Structure

Byte 0: Command byte

0x84 = Change hardware filter

Byte 1: Success/Failure

0 = Failure

1 = Success

The set hardware filters command is not yet functional. All filtering is currently being done by software.

## 13.8 Change Cyclic Tx Message

### Command Message Structure

Byte 0: Command byte

0x05 = Change cyclic Tx message

Byte 1: CAN port

0 = CAN port 1

1 = CAN port 2

Bytes 2 – 5: Broadcast Rate (ms)

Bytes 6 – 9: 11 or 29 bit ID

Bytes 10 – 11: DLC (data length code) of the message. This number cannot be greater than 8

Bytes 12 – n: Body of the message.

### Response Message Structure

Byte 0: Command byte

0x85 = Change cyclic Tx message

Byte 1: Success/Failure

0 = Failure

1 = Success

## 13.9 Remove Cyclic Tx Message

### Command Message Structure

Byte 0: Command byte

0x06 = Remove cyclic Tx message

Bytes 1 – 4: 11 or 29 bit ID

Byte 5: CAN port

0 = CAN port 1

1 = CAN port 2

### Response Message Structure

Byte 0: Command byte

0x86 = Remove cyclic Tx message

Byte 1: Success/Failure

0 = Failure

1 = Success

## 13.10 Change Baud Rate Message

Change the baud rate and ID type for CAN ports 1 & 2.

### Command Message Structure

Byte 0: Command byte

0x07 = Change baud rate message command byte

Byte 1: CAN port 1 ID type,

0 = Standard CAN ID, 1 = Extended CAN ID

Byte 2: CAN port 1 baud rate

0 = 10k

1 = 20k

2 = 33.33k

3 = 50k

4 = 62.5k

5 = 75k  
6 = 83.3k  
7 = 100k  
8 = 125k  
9 = 250k  
10 = 500k  
11 = 800k  
12 = 1M

Byte 3: CAN port 2 ID type,

0 = Standard CAN ID, 1 = Extended CAN ID

Byte 4: CAN port 2 baud rate

0 = 10k  
1 = 20k  
2 = 33.33k  
3 = 50k  
4 = 62.5k  
5 = 75k  
6 = 83.3k  
7 = 100k  
8 = 125k  
9 = 250k  
10 = 500k  
11 = 800k  
12 = 1M

#### Response Message Structure

Byte 0: Command byte

0x87 = Change baud rate

Byte 1: Success/Failure

0 = Failure  
1 = Success

## 13.11 Get Baud Rate Message

Retrieves the baud rate settings for CAN ports 1 and 2.

#### Command Message Structure

Byte 0: Command byte

0x08= Change baud rate message command byte

#### Response Message Structure

Byte 0: Command byte

0x88 = Change baud rate message command byte

Byte 1: CAN port 1 ID type,

0 = Standard CAN ID, 1 = Extended CAN ID

Byte 2: CAN port 1 baud rate

0 = 10k  
1 = 20k  
2 = 33.33k  
3 = 50k  
4 = 62.5k  
5 = 75k  
6 = 83.3k  
7 = 100k  
8 = 125k  
9 = 250k



10 = 500k

11 = 800k

12 = 1M

Byte 3: CAN port 2 ID type,

0 = Standard CAN ID, 1 = Extended CAN ID

Byte 4: CAN port 2 baud rate

0 = 10k

1 = 20k

2 = 33.33k

3 = 50k

4 = 62.5k

5 = 75k

6 = 83.3k

7 = 100k

8 = 125k

9 = 250k

10 = 500k

11 = 800k

12 = 1M

## 13.12 Initialize Log Payback Timer Message

This command is used if CAN Tx messages, using the delay time, are loaded into the Translator. This command initializes the beginning of the delay time to the current stack timer. As Tx messages are loaded into the Translator for broadcast, the Tx messages delay time is added to the initialized delay time to result in a future time when the Tx message is to be broadcast.

### Command Message Structure

Byte 0: Command byte

0x09 = Change baud rate message command byte

### Response Message Structure

None

## 13.13 Get Message Converter Data Message

### Command Message Structure

Byte 0: Command byte

0x0A = Get message converter data

### Response Message Structure

Byte 0: Command byte

0x8A = Get message converter data

Byte 1: Number of conversion parameter

0 – 20 = This is the range or max number of conversion parameter that can be sent to the Translator

Byte 2: Converter parameter enabled

0 = FALSE

1 = TRUE

Byte 3: CAN port

0 = CAN port 1

1 = CAN port 2

Byte 4: Rx transport type

0 = Single packet

1 = ISO multipacket

2 = NMEA fastpacket

Byte 5: Tx transport type

- 0 = Single packet
- 1 = ISO multipacket
- 2 = NMEA fastpacket

Byte 6: Rx data type

- 0 = Unsigned
- 1 = Signed
- 2 = Float

Byte 7: Tx data type

- 0 = Unsigned
- 1 = Signed
- 2 = Float

Byte 8: Rx start bit

- 0 – 8 = Bit number within the start byte where the parameter starts.

Byte 9: Tx start bit

- 0 – 8 = Bit number within the start byte where the parameter starts.

Bytes 10 – 13: ID of the received message

Bytes 14 – 17: ID mask for the Rx ID

Bytes 18 – 21: ID of the transmitted message

Bytes 22 – 23: Rx parameter start byte

Bytes 24 – 25: Tx parameter start byte

Bytes 26 – 27: Rx parameter bit length

Bytes 28 – 29: Tx parameter bit length

Bytes 30 – 33: Rx parameter resolution (float)

Bytes 34 – 37: Tx parameter resolution (float)

Bytes 38 – 41: Rx parameter offset

Bytes 42 – 45: Tx parameter offset

Bytes 46 – 49: Tx message broadcast rate (ms)

Bytes 50 – 53: Tx message broadcast count

Bytes 54 – 57: Tx message DLC

Repeat bytes 2 – 57 for each of the parameter conversions that need to be added to the stack. The number of conversions cannot exceed 20.

## 13.14 Change Message Converter Data Message

### Command Message Structure

Byte 0: Command byte

- 0x0B = Change message converter data

Byte 1: Number of conversion parameter

- 0 – 20 = This is the range or max number of conversion parameter that can be sent to the Translator

Byte 2: Converter parameter enabled

- 0 = FALSE
- 1 = TRUE

Byte 3: CAN port

- 0 = CAN port 1
- 1 = CAN port 2

Byte 4: Rx transport type

- 0 = Single packet
- 1 = ISO multipacket
- 2 = NMEA fastpacket

Byte 5: Tx transport type

- 0 = Single packet
- 1 = ISO multipacket
- 2 = NMEA fastpacket

Byte 6: Rx data type

- 0 = Unsigned
- 1 = Signed
- 2 = Float

Byte 7: Tx data type

- 0 = Unsigned
- 1 = Signed
- 2 = Float

Byte 8: Rx start bit

- 0 – 8 = Bit number within the start byte where the parameter starts.

Byte 9: Tx start bit

- 0 – 8 = Bit number within the start byte where the parameter starts.

Bytes 10 – 13: ID of the received message

Bytes 14 – 17: ID mask for the Rx ID

Bytes 18 – 21: ID of the transmitted message

Bytes 22 – 23: Rx parameter start byte

Bytes 24 – 25: Tx parameter start byte

Bytes 26 – 27: Rx parameter bit length

Bytes 28 – 29: Tx parameter bit length

Bytes 30 – 33: Rx parameter resolution (float)

Bytes 34 – 37: Tx parameter resolution (float)

Bytes 38 – 41: Rx parameter offset

Bytes 42 – 45: Tx parameter offset

Bytes 46 – 49: Tx message broadcast rate (ms)

Bytes 50 – 53: Tx message broadcast count

Bytes 54 – 57: Tx message DLC

Repeat bytes 2 – 57 for each of the parameter conversions that need to be added to the stack. The number of conversions cannot exceed 20.

#### Response Message Structure

Byte 0: Command byte

- 0x8B = Change message converter data

Byte 1: Success/Failure

- 0 = Failure
- 1 = Success

## 13.15 Get Translator Timer Tick

#### Command Message Structure

Byte 0: Command byte

- 0x0C = Get Translator timer tick

#### Response Message Structure

Byte 0: Command byte

- 0x8C = Get Translator timer tick

Bytes 1 – 8: 64 bit Translator timer tick

## 13.16 Clear USB Tx Buffer

This command clears the Translators USB Tx buffer. This must be done after a PC connects to the Translator via USB if there has been CAN bus traffic on either CAN port. CAN traffic causes the Translator to try to Tx a message to the PC. If a PC is not attached the USB Tx interrupt cannot occur and will not occur even after the PC is attached. Thus this command should be sent to the Translator by the PC after a connection has been established.

Command Message Structure

Byte 0: Command byte

0x0E = Clear USB Tx buffer

Response Message Structure

Byte 0: Command byte

0x8E = Clear USB Tx buffer

Byte 1: Success/Failure

0 = Failure

1 = Success

## 13.17 Set Translator Timer Tick

Command Message Structure

Byte 0: Command byte

0x0D = Set Translator timer tick

Response Message Structure

Byte 0: Command byte

0x8D = Set Translator timer tick

Byte 1: Success/Failure

0 = Failure

1 = Success

## 13.18 Get Translator Settings Message

This command retrieves a general list of settings from the Translator.

These settings include: Baud rate, gateway, SD Card logging and filters

Command Message Structure

Byte 0: Command byte

0x17 = Get Translator settings

Response Message Structure

Byte 0: Command byte

0x97 = Change baud rate message command byte

Byte 1: CAN port 1 ID type,

0 = Standard CAN ID, 1 = Extended CAN ID

Byte 2: CAN port 1 baud rate

0 = 10k

1 = 20k

2 = 33.33k

3 = 50k

4 = 62.5k

5 = 75k

6 = 83.3k

7 = 100k

8 = 125k

9 = 250k

10 = 500k

11 = 800k

12 = 1M

Byte 3: CAN port 2 ID type,

0 = Standard CAN ID, 1 = Extended CAN ID

Byte 4: CAN port 2 baud rate

- 0 = 10k
- 1 = 20k
- 2 = 33.33k
- 3 = 50k
- 4 = 62.5k
- 5 = 75k
- 6 = 83.3k
- 7 = 100k
- 8 = 125k
- 9 = 250k
- 10 = 500k
- 11 = 800k
- 12 = 1M

Byte 5: Gateway enable for CAN port 2 to CAN port 1

- 0 = FALSE
- 1 = TRUE

Byte 6: Gateway active before/after filtering on CAN port 2

- 0 = Before filtering
- 1 = After filtering

Byte 7: Gateway enable for CAN port 1 to CAN port 2

- 0 = FALSE
- 1 = TRUE

Byte 8: Gateway active before/after filtering on CAN port 1

- 0 = Before filtering
- 1 = After filtering

Byte 9 – 12: SD Card log file write time. This is the amount of time (ms) that the Translator will cache received CAN messages before writing them to the SD Card. This time is reset if a write occurs prior to timeout. Refer to section 6.1.3.4.

Byte 13 – 16: SD Card log file size. This is the max size the SD Card log file can be. When the current file size reaches the set point, the file is closed and a new file is started. Refer to section 6.1.3.5.

Byte 17: SD Card logging active

- 0 = FALSE;
- 1 = TRUE;

Byte 18: Log file type

- 0 = ASCII Decimal
- 1 = ASCII Hexadecimal
- 2 = National Instruments file format

Byte 19: Include Translator Time Tick in the log file

- 0 = FALSE;
- 1 = TRUE;

Byte 20: CAN port to apply filter data to

- 0 = CAN port 1
- 1 = CAN port 2

Byte 21: Filters active

- 0 = FALSE;
- 1 = TRUE;

Byte 22: Number of filters data to follow

Bytes 23 – 26: 11 or 29 bit filter

Bytes 27 – 30: 11 or 29 bit mask

Byte 31: Flag indicating if filter is inclusive/exclusive

- 0 = Inclusive
- 1 = Exclusive

Byte 32: Filter Byte; Byte within the body of the message to apply the byte filter

Byte 33: Byte filter

Byte 34: Byte mask

Repeat bytes 26 – 34 for the number of filters applied to the CAN port ID defined in Byte 18.  
Repeat bytes 20 – 34 for the opposite CAN port.

## 13.19 Save Filters Message

### Command Message Structure

Byte 0: Command byte

0x18 = Change baud rate message command byte

Byte 1: CAN port 1 ID type,

0 = Standard CAN ID, 1 = Extended CAN ID

Byte 2: CAN port 1 baud rate

0 = 10k

1 = 20k

2 = 33.33k

3 = 50k

4 = 62.5k

5 = 75k

6 = 83.3k

7 = 100k

8 = 125k

9 = 250k

10 = 500k

11 = 800k

12 = 1M

Byte 3: CAN port 2 ID type,

0 = Standard CAN ID, 1 = Extended CAN ID

Byte 4: CAN port 2 baud rate

0 = 10k

1 = 20k

2 = 33.33k

3 = 50k

4 = 62.5k

5 = 75k

6 = 83.3k

7 = 100k

8 = 125k

9 = 250k

10 = 500k

11 = 800k

12 = 1M

Byte 5: CAN port to apply filter data to

0 = CAN port 1

1 = CAN port 2

Byte 6: Filters active

0 = FALSE;

1 = TRUE;

Byte 7: Number of filters data to follow

Bytes 8 – 11: 11 or 29 bit filter

Bytes 12 – 15: 11 or 29 bit mask

Byte 16: Flag indicating if filter is inclusive/exclusive

0 = Inclusive

1 = Exclusive

Byte 17: Filter Byte; Byte within the body of the message to apply the byte filter

Byte 18: Byte filter

Byte 19: Byte mask

Repeat bytes 8 – 19 for the number of filters applied to the CAN port ID defined in Byte 18.  
Repeat bytes 5 – 19 for the opposite CAN port.

#### Response Message Structure

Byte 0: Command byte

0x98 = Change baud rate message command byte

Byte 1: Success/Failure

0 = Failure

1 = Success

## 13.20 Save SD Card Log File Settings Message

#### Command Message Structure

Byte 0: Command byte

0x19 = Save SD card log file settings

Bytes 1 – 4: Log file save time

Bytes 5 – 8: Max log file size

Byte 9: SD card file logging active

0 = FALSE

1 = TRUE

#### Response Message Structure

Byte 0: Command byte

0x99 = Save SD card log file settings received

Byte 1: Success/Failure

0 = Failure

1 = Success

## 13.21 Ready To Rx CAN Messages Flag Message

#### Command Message Structure

Byte 0: Command byte

0x1A = Ready to Rx CAN message flag

#### Response Message Structure

Byte 0: Command byte

0x9A = Ready to Rx CAN messages flag received

Byte 1: Success

0 = Failure

1 = Success

## 13.22 Change Gateway Settings Message

#### Command Message Structure

Byte 0: Command byte

0x1B = Change gateway settings

Byte 1: Gateway enable for CAN port 2 to CAN port 1

0 = FALSE

1 = TRUE

Byte 2: Gateway active before/after filtering on CAN port 2

0 = Before filtering

1 = After filtering

Byte 3: Gateway enable for CAN port 1 to CAN port 2

0 = FALSE

1 = TRUE

Byte 4: Gateway active before/after filtering on CAN port 1

0 = Before filtering

1 = After filtering

#### Response Message Structure

Byte 0: Command byte

0x9B = Change gateway settings

Byte 1: Success/Failure

0 = Failure

1 = Success

## **13.23 Format SD Card Message**

#### Command Message Structure

Byte 0: Command byte

0x1C = Format SD Card message

#### Response Message Structure

Byte 0: Command byte

0x9C = Format SD Card message

Byte 1: Success/Failure

0 = Failure

1 = Success

## **13.24 Get Revision Information Message**

#### Command Message Structure

Byte 0: Command byte

0x20 = Get Revision Information Message

#### Response Message Structure

Byte 0: Command byte

0xA0 = Get Revision Information Message

Bytes 1 – 2: Hardware version (LSB, BCD, Max 255.255)

Bytes 3 – 4: Boot version (LSB, BCD, Max 255.255)

Bytes 5 – 6: Application version (LSB, BCD, Max 255.255)

Bytes 7 – 10: Serial number