SILICON ENGINES

CAN/USB CONVERTER
MODEL 9012

USER'S GUIDE

WINDOWS® COMPATIBILITY

The Model 9012 is compatible with Windows 2000, Windows XP, Windows Vista, and Windows 7

COMMENTS

We would appreciate receiving corrections and suggestions regarding this document and the product it describes. Please email to sales@siliconengines.net

SILICON ENGINES

3550 W Salt Creek Lane
Suite 105
Arlington Heights, IL 60005 USA
847-637-1180
FAX 847-637-1185
www.siliconengines.net

CONTENTS

1. INTRODUCTION 4
1.1. SCOPE 4
1.2. RELATED AUTOMOTIVE PROTOCOL CONVERTERS 4
1.3. MODEL 9012 FUNCTIONS 4
-2. HARDWARE OUTLINE 5
2. HARDWARE OUTLINE 6
2.1. ENCLOSURE 6
2.2. ENCLOSURE SIZE 6
2.3. TOP PANEL 7
2.4. CAN CONNECTOR 7
2.5. INDUSTRY STANDARD DB9 PINOUT 8
2.6. CAN CABLE 8
2.7. CAN BUS TERMINATION 8
2.8. USB CONNECTOR 9
3. POWER REQUIREMENTS 9
3.1. USB POWER 9
3.2. ISOLATION 9
3.3. CAN POWER 9
4. LAMP FUNCTIONS 10
4.1. USB LAMP 10
4.2. CAN LAMP 10
4.3. STATUS LAMP 10
5. CAN/USB MESSAGE CENTER—INSTALLATION 11
5.1. FUNCTIONALITY 11
5.2. INSTALLATION 11
5.3. INSTALLATION OF USB DEVICE DRIVERS 11
5.4. WINDOWS 7 AND VISTA INSTALLATION SCREENS 12
5.5. WINDOWS XP AND 2000 INSTALLATION SCREENS 13
6. CAN USB MESSAGE CENTER 16
6.1. PROVIDED PC SOFTWARE 16
6.2. MAIN MESSAGE SCREEN 16
6.3. CONFIGURATION SCREEN 17
6.4. ADDING CAN NODES 18
6.5. EDITING CAN MESSAGES 19
6.6. ADDING A NEW CAN MESSAGE 19
6.7. SENDING A CAN MESSAGE 20
6.8. SENDING A CAN MESSAGE REPEATEDLY 20
7. TRIGGER OUTPUT SIGNAL 20
7.1. TRIG OUTPUT, PIN 8 ON CAN CONNECTOR 20
7.2. TRIGGER SIGNAL DESCRIPTION 20
7.3. TRIGGER SIGNAL LOADING 20
8. BACKGROUND ON THE CAN BUS 21
8.1. OUTLINE SPECIFICATIONS 21
8.2. RELATED PROTOCOLS 21
8.3. PHYSICAL LAYER 21
8.4. CAN CIRCUITRY WITHIN EACH ECU 21
8.5. SERIAL DATA SPEED 22
9. MODEL 9012 HARDWARE 22
9.1. BLOCK DIAGRAM 22
9.2. HARDWARE DESCRIPTION 22
10. EMBEDDED SOFTWARE OUTLINE 23
10.1. CAN FEATURES 23
10.2. FIRMWARE UPGRADES 23
11. REVISION HISTORY 23
11.1. REVISION A 23

1. INTRODUCTION

1.1. SCOPE

This document is the User's Guide for the Silicon Engines ***Model 9012 CAN/USB Converter,*** a compact electronic device that allows a personal computer to connect to a dual wire CAN network.

1.2. RELATED AUTOMOTIVE PROTOCOL CONVERTERS

In addition to the Model 9012 LIN/USB Converter, Silicon Engines also offers:

- **Model 9001**, 9141/RS-232 Converter
- **Model 9002**, 14230/RS-232 Converter
- **Model 9003,** LIN/RS-232 Converter
- **Model 9010**, 14230/USB Converter.
- **Model 9011**, LIN/USB Converter.

For details on these devices, check our website, www.siliconengines.net.

1.3. MODEL 9012 FUNCTIONS

- **Protocol conversion:** Provides a robust and complete conversion between the CAN (Controller Area Network) protocol and USB (Universal Serial Bus), for connection to a personal computer.
- **Physical layer conversion:** CAN and USB transceivers connect to a dedicated microcontroller within the Model 9012 to convert between the two protocols.
- **CAN data speeds:** Programmable for a wide variety of CAN bus data speeds from 1,000 bps (bits per second) to 1 Mbps (1 megabit per second) over the CAN bus. This range includes the standard speeds recommended in the CAN specifications—100,000, 250,000, 500,000 and of course 1,000,000 bps. The data rate can also be set to any non-standard speed within the covered range to test the ability of slave mode ECUs to synchronize with master ECUs. Also, the Model 9012 can be queried as to the rate at which the bus is currently configured so that the developer can ensure that the correct bus speed is being produced.
- **CAN bit sampling parameters:** The Model 9012 is programmable as to the timing of the bit sampling mechanism within the CAN bus receiver**.**
- **Programmable CAN bus termination:** Able to configure whether or not the CAN bus termination resistor within the Model 9012 is turned on.
- **Operating modes:** Able to configure a test mode with an automatic incrementing payload; able to configure whether there is automatic recovery from bus-off.
- **Timed transmissions:** Can be set up to generate timed transmissions of messages, completely asynchronously from a host controller.
- **Always listening:** Will listen to all CAN traffic at the specified bus speed on the bus, and report all traffic to the PC.
- **USB data speed:** Operates at USB 2.0 high speed, 480 megabits per second raw data rate over the USB bus.

- **Supplied PC software:** A complete Visual Basic® 6.0 PC software program—operating under Microsoft® Windows®—is included with the CAN/USB Converter, with source code provided. Utilities are provided for configuring the CAN/USB Converter; sending and receiving CAN bus messages; and logging CAN message traffic to a file on the PC. There are reusable header files as well, for both Visual C++ 6.0 and full source code written in Visual Basic 6.0 and Visual Basic .NET 2008, intended to be used with the CANUSB.DLL or CANUSB64.DLL (for true 64-bit processing if needed) that handles the interface to the CAN/USB Converter.

  The Model 9012 is compatible with Windows 2000, Windows XP, Windows Vista, and Windows 7.

- **Multiple port capability:** The PC software handles multiple connected CAN/USB Converters simultaneously through a single user interface. A typical arrangement might be one USB master, two USB slaves, and a device operating in PC control mode. Hot-swapping of CAN/USB Converters is permitted.
- **Stop on Error:** In addition, the CAN/USB Converter can be programmed to ***stop on error,*** useful in debugging CAN communications problems.
- **Output Trigger:** The Model 9012 CAN/USB Converter can be programmed to assert a pulse on its TRIGGER output upon command, or after a specific microsecond delay from the start of a particular CAN message with a particular ID. This can be useful for triggering other devices or triggering an oscilloscope.
- **Windows® DLL:** An easy-to-use DLL file is provided to allow the user to write custom CAN interfaces. Like the provided software program, the DLL supports multiple connected CAN/USB Converters. There is also a 64-bit DLL provided for 64-bit application support.

-

2. HARDWARE OUTLINE

2.1. ENCLOSURE

The LIN/USB Converter is housed in a compact plastic enclosure.

MODEL 9012 ENCLOSURE
FIGURE 2.1.1

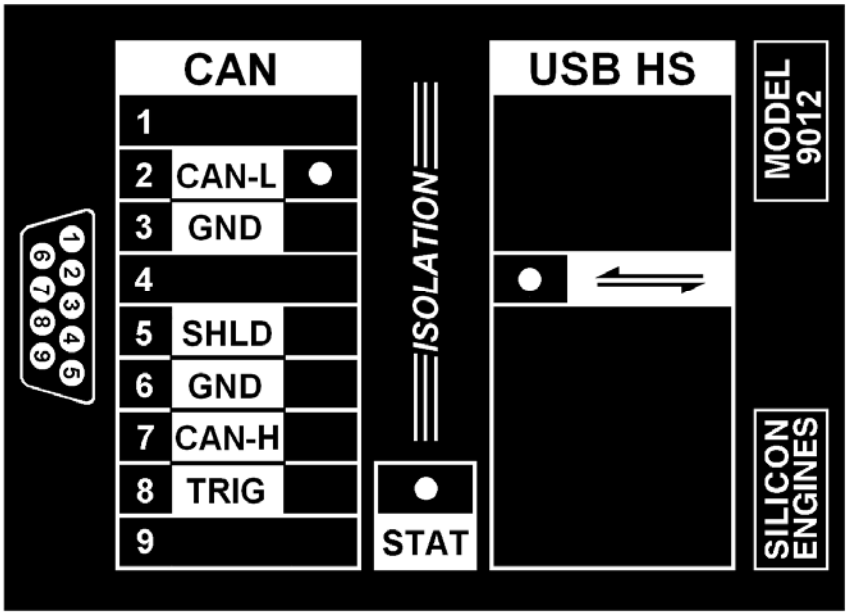
2.2. ENCLOSURE SIZE

WIDTH	HEIGHT	DEPTH
4.375 IN	3.25 IN	1.5 IN
111 MM	82,6 MM	38,1 MM

ENCLOSURE DIMENSIONS
FIGURE 2.2.1.

2.3. TOP PANEL

Here is a close-up view of the graphic overlay on the top panel of the enclosure.



TOP PANEL OF ENCLOSURE
FIGURE 2.3.1

2.4. CAN CONNECTOR

The connector at the left of the Model 9012 LIN/USB Converter is a type DB9M plug (9-pin male D sub-miniature). A cable with a DB9F (female DB9) socket plugs in here. Here is a description of signals on these pins:

PIN NO.	SYMBOL	DESCRIPTION
1		RESERVED, DO NOT USE
2	**CAN-L**	CAN BUS LOW SIGNAL LINE
3	**GND**	CAN BUS SIGNAL GROUND
4		RESERVED, DO NOT USE
5	**SHLD**	OPTIONAL CAN BUS CABLE SHIELD CONNECTION
6	**GND**	CAN BUS SIGNAL GROUND
7	**CAN-H**	CAN BUS HIGH SIGNAL LINE
8	**TRIG**	TRIGGER OUTPUT
9		RESERVED, DO NOT USE

CAN CONNECTOR PIN-OUTS
FIGURE 2.4.1.

These pin-outs, as well as the locations of the pins within the 9-pin connector, are shown on the Model 9012 top panel *(Fig. 2.3.1).*

2.5. INDUSTRY STANDARD DB9 PINOUT

The pin assignments on the Model 9012 DB9 connector are intended to be consistent with de facto industry standards. See for example the CANopen specifications at http://www.can-cia.org.

However users should carefully check the documentation for other CAN bus equipment to assure compatibility with the Model 9012.

2.6. CAN CABLE

Users must construct a special cable to connect the Model 9012 CAN/USB Converter to the CAN bus being monitored, or the ECU being tested. Or contact Silicon Engines for assistance.

At the Model 9012 side, use a DB9F connector. Minimum connections include:

PIN NO.	SYMBOL	DESCRIPTION
2	**CAN-L**	CAN BUS LOW SIGNAL LINE
3	**GND**	CAN BUS SIGNAL GROUND
7	**CAN-H**	CAN BUS HIGH SIGNAL LINE

MINIMUM CAN CONNECTIONS
FIGURE 2.6.1.

The following additional connections are optional, but may be useful when dealing with higher CAN data rates, long bus lines, or especially noisy electrical environments:

PIN NO.	SYMBOL	DESCRIPTION
5	**SHIELD**	CONNECT TO SHIELD ON SHIELDED CAN BUS CABLE
6	**GND**	CONNECT IN PARALLEL WITH PIN 3

OPTIONAL CAN CONNECTIONS
FIGURE 2.6.2.

The following terminals are reserved for future Silicon Engines products, and should not be connected:

PIN NO.	DESCRIPTION
1	RESERVED, DO NOT CONNECT
4	RESERVED, DO NOT CONNECT
9	RESERVED, DO NOT CONNECT

RESERVED TERMINALS, DO NOT USE
FIGURE 2.6.3.

2.7. CAN BUS TERMINATION

The Model 9012 has a built-in bus termination resistor that can be turned on or off under software control.

PARAMETER	MINIMUM	TYPICAL	MAXIMUM	COMMENTS
CAN BUS TERMINATION RESISTOR	116.4Ω	120Ω	123.6Ω	120Ω ±3%

CAN BUS TERMINATION RESISTOR
FIGURE 2.6.1

2.8. USB CONNECTOR

The connector at the right side of the Model 9012 CAN/USB Converter is a USB type B female receptacle. It is intended for connection via a USB cable to a USB port on a personal computer.

A suitable USB cable is provided with each Model 9012, measuring six feet (1,8 meters) in length.

USB CABLE FROM MODEL 9012 TO PERSONAL COMPUTER
FIGURE 2.8.1.

At one end of the cable is a USB type B male plug. The type B plug is shown at the left in the illustration above. It plugs into the USB type B receptacle on the Model 9012.

At the other end of the cable is a USB type A male plug. This type A plug is shown at the right in the illustration above. It plugs into a USB port on the PC.

3. POWER REQUIREMENTS

3.1. USB POWER

The Model 9012 CAN/USB Converter contains built-in power supply circuitry that generates all needed power from the USB port into which the device is plugged.

3.2. ISOLATION

The circuitry on the USB side of the Model 9012 is galvanically isolated from the circuitry on the CAN side of the Model 9012. This permits usage of this product when there are significant ground differentials between the CAN bus system, and the signal grounds present on the personal computer.

3.3. CAN POWER

Power for the CAN transceiver is generated by an isolated voltage regulator within the Model 9012 that derives its power from the USB supply. There is no need to provide a power voltage (for example, 12V) on the CAN side of the system.

4. LAMP FUNCTIONS

4.1. USB LAMP

At the right of the top panel of the Model 9012 CAN/USB Converter is a bi-color LED lamp, marked **USB**.

LED PATTERN	CONDITION
STEADY GREEN	THIS DEVICE ENUMERATED BY PC USB SOFTWARE, CONVERTER OPERATING NORMALLY
GREEN/RED BLINKING	USB TRAFFIC DETECTED
LONG RED, SHORT GREEN BLINKING	THIS DEVICE NOT YET ENUMERATED BY PC
STEADY RED	HARDWARE ERROR

USB LAMP PATTERNS
FIGURE 4.1.1.

4.2. CAN LAMP

At the left of the top panel of the CAN/USB Converter is a bi-color LED lamp, marked **CAN**.

LED PATTERN	CONDITION
STEADY GREEN	CONVERTER OPERATING NORMALLY, CAN BUS IDLE
GREEN/RED BLINKING	CAN TRAFFIC DETECTED
STEADY RED	CAN BUS ERROR

CAN LAMP PATTERNS
FIGURE 4.2.1.

4.3. STATUS LAMP

At bottom center on the top panel of the CAN/USB Converter is a green LED lamp, marked **STAT**.

LED PATTERN	CONDITION
STEADY GREEN	CONVERTER OPERATING NORMALLY
GREEN BLINKING	NEW SOFTWARE BEING DOWNLOADED. OTHERWISE, HARDWARE ERROR

LIN LAMP PATTERNS
FIGURE 4.3.1.

5. CAN/USB MESSAGE CENTER—INSTALLATION

5.1. FUNCTIONALITY

The **CAN USB Message Center** software utility is provided with the Model 9012 CAN/USB Converter. When installed in a personal computer running Windows®, this program allows you to send and receive messages through the CAN/USB Converter to a connected ECU or other CAN-enabled device.

The Model 9012 is compatible with Windows 2000, Windows XP, Windows Vista, and Windows 7.

5.2. INSTALLATION

The **CAN USB Message Center** software is provided on a CD that comes with the product. This software is also available on the Silicon Engines website. To install the software:

1. Insert the CD into your computer's CD drive. Find the following program options on the root directory of the CD.

2. For a 32-bit computer, run the program "Install SE 9012 CANUSB Message Center [xxxx] and Drivers [xxxx] 32-Bit.exe", where [xxxx] is the latest software version.

3. For a 64-bit computer, run "Install SE 9012 CANUSB Message Center [xxxx] and Drivers [xxxx] 64-Bit.exe", where [xxxx] is the latest version

4. The selected program will install the software on your computer.

5. If you want to install just the drivers and not the Message Center then run either the program "Install 32-Bit SE 9012 9013 and 9014 Drivers [xxxx].exe" or "Install 64-Bit SE 9012 9013 and 9014 Drivers [xxxx].exe" (where [xxxx] is the latest version).

Please install the Message Center software first before plugging in the CAN/USB Converter for the first time.

5.3. INSTALLATION OF USB DEVICE DRIVERS

The USB device drivers are installed along with whichever one of the four executables in the root folder of the CD that you have chosen. The CD that comes with the product also contains in the Drivers folder the files necessary for USB initialization. These files include:

1. In the directory `C:\Program Files (x86)\Silicon Engines\CANUSB Driver x32 on x64\drivers J10.21 9012 9013 9014 x32 on x64 (for 64-bit computers when just the drivers were installed),`

2. Alternately `C:\Program Files (x86)\Silicon Engines\Silicon Engines 9012 CAN USB Message Center 64-Bit\drivers J10.21 9012 9013 9014 x32 on x64` (for 64-bit computers when the message center was installed),

3. Alternately `C:\Program Files\Silicon Engines\CANUSB Drivers x32\drivers J10.21 9012 9013 9014 x32` (for 32-bit computers when just the drivers were installed)

4. Alternately `C:\Program Files\Silicon Engines\Silicon Engines 9012 CAN USB Message Center\drivers J10.21 9012 9013 9014 x32` (for 32-bit computers when the message center was installed).

Here are the steps necessary to install the USB device drivers:

Message Center: Install the CAN/USB Message Center software into your PC.

USB: Connect the USB cable from the Model 9012 to your PC.

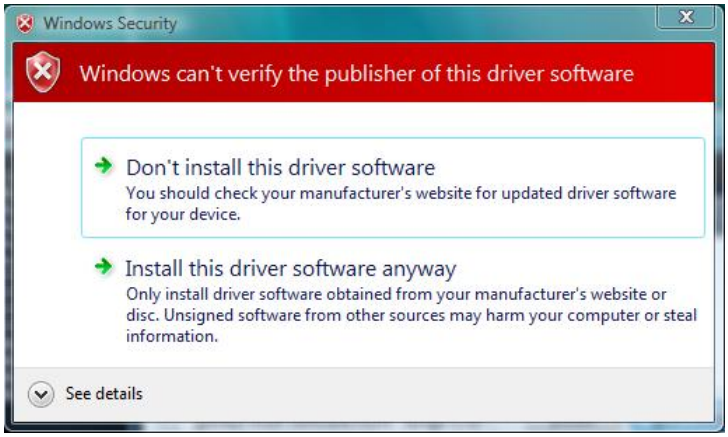
5.4. WINDOWS 7 AND VISTA INSTALLATION SCREENS

If you are running Windows 7 or Vista, you should see a screen as shown below.

DEVICE SOFTWARE INSTALL SCREEN
FIGURE 5.4.1

Note: Jungo is a supplier of PC-side USB software utilities which Silicon Engines uses for the CAN/USB Message Center.

Go ahead and click the INSTALL button. You should then see the following screen:



WINDOWS PUBLISHER CHECK
FIGURE 5.4.2

Please click on INSTALL THIS DRIVER SOFTWARE ANYWAY.

The driver software for Windows 7 or Windows Vista should now be installed.

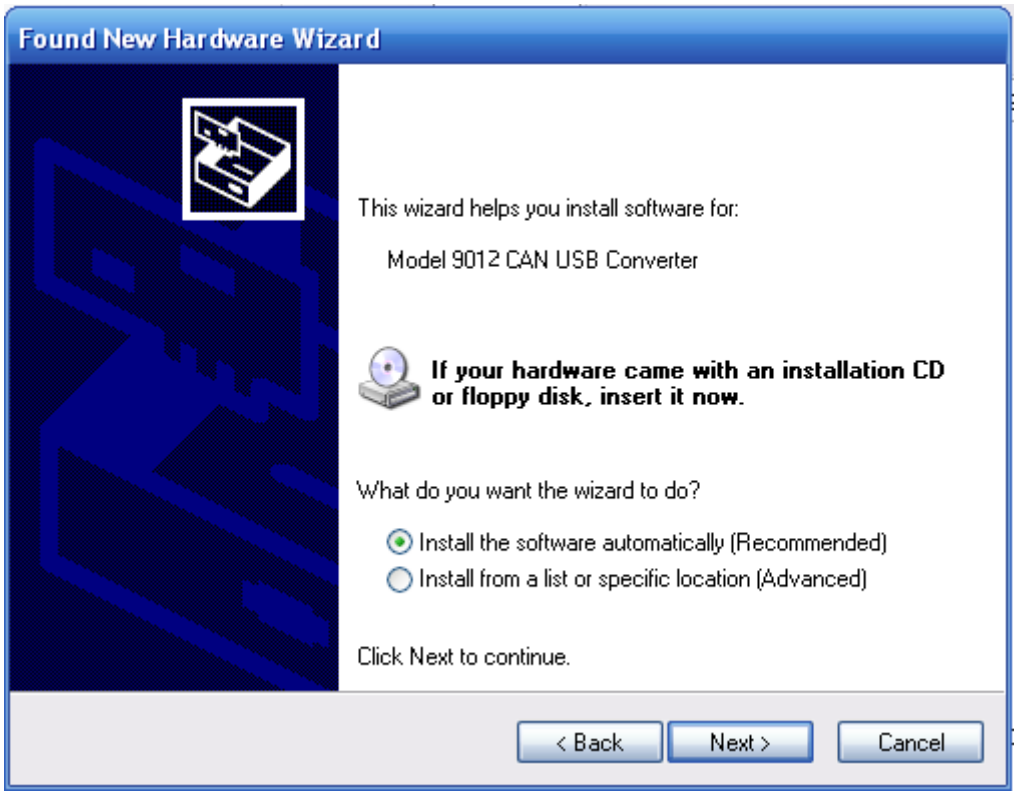
5.5. WINDOWS XP AND 2000 INSTALLATION SCREENS

If you are running Windows XP or 2000 you should see a screen as shown below.

WINDOWS XP AND 2000 INSTALLATION SCREEN
FIGURE 5.5.1

Go ahead and click the NO, NOT THIS TIME button.

You should then see the following screen:



WINDOWS XP/2000 INSTALL SCREEN #2
FIGURE 5.5.2

Please click on INSTALL THIS SOFTWARE AUTOMATICALLY.

You should next see the following screen:

WINDOWS XP/2000 INSTALL SCREEN #3
FIGURE 5.5.3

Please click the FINISH button. The software should now be installed on your computer.

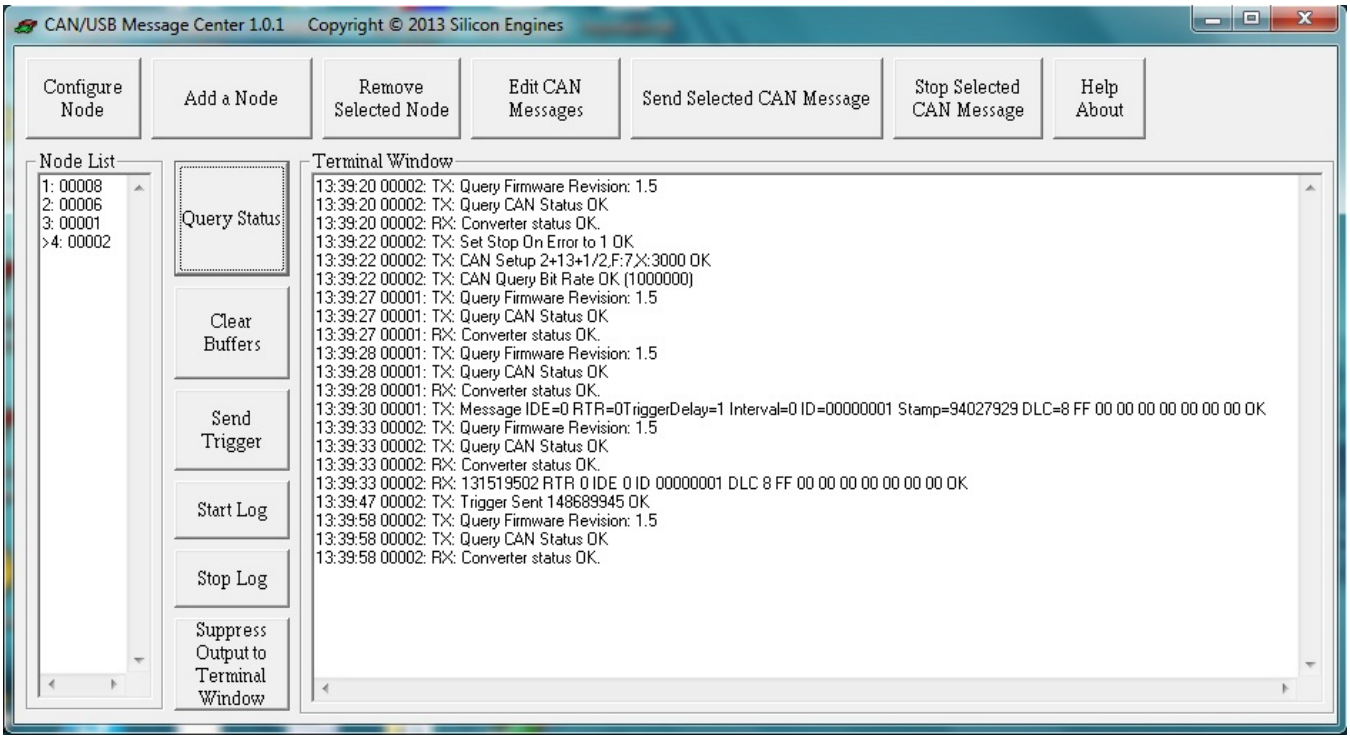
6. CAN USB MESSAGE CENTER

6.1. PROVIDED PC SOFTWARE

The CAN USB Message Center is a PC-based program for controlling one or more CAN/USB Converters. This program is provided at no additional cost with the device. Full source code for the CAN USB Message Center is also provided so that users can build their own applications with the least amount of learning curve.

6.2. MAIN MESSAGE SCREEN

The Main Message Screen is the top-level control center for the Message Center program.



CAN/USB MESSAGE CENTER MAIN MESSAGE SCREEN
FIGURE 6.2.1

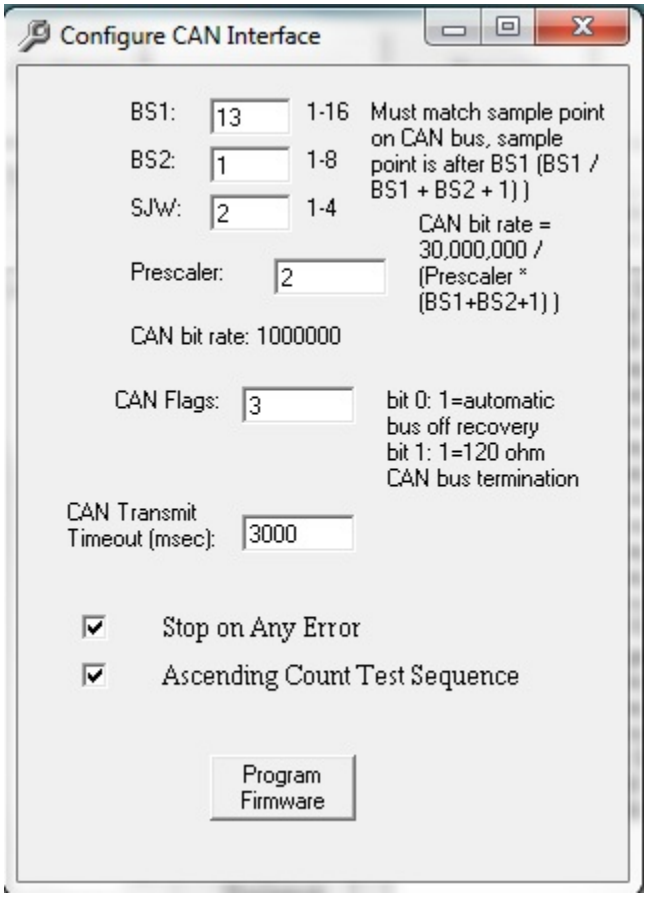
- **Terminal Window:** An active scrolling window showing all messages to and from CAN/USB Converters. Each line contains a time stamp, node number, transmit/receive indication, and message description.
- **Configure CAN Box:** Sets up attached CAN/USB Converters.
- **Node List:** The ***Node List*** window shows the currently active CAN nodes (model 9012 devices shown only).
- **Node Control:** The ***Add a Node*** button allows you to add and identify additional CAN/USB Converters. Use ***Add a Node*** to add a node that does not appear in the Node List to the Node list to be available for Selecting. To select a node, simply place the mouse cursor over the Node's serial number in the Node List and click the mouse button. Use the ***Remove Selected Node*** button to remove the node from the list.
- **Other CAN nodes:** Nodes on the CAN bus that are not Model 9012, 9013, or 9014 CAN/USB Converters—CAN-compatible automotive ECUs—will not appear in the ***Node List***. However their message traffic will appear in the ***Terminal Window***.

- **Message Control:** The ***Edit CAN Messages***, ***Send Selected CAN Message***, and ***Stop Selected CAN Message*** buttons are used to define CAN messages, and to send them, once or repeatedly, or to stop them from being sent.
- **Query Status:** Used to cause a connected CAN/USB Converter to report its current firmware version and operating status, including any existing error messages.
- **Clear Buffers:** Use this button to clear the receive buffers and also to clear any error conditions.
- **Send Trigger:** Use this button to send a trigger signal on the TRIGGER pin and record the microsecond timestamp when the trigger action occurred.

  (See Part 7 for the electrical specifications for the TRIGGER output signal.)
- **Buffer control:** The ***Start Log*** button causes all bus activity to be logged to a designated PC file. The ***Stop Log*** button stops data logging. ***Suppress Output to Terminal Window*** will freeze the messages in the terminal window, while allowing other CAN activities to continue in the background.

6.3. CONFIGURATION SCREEN

This screen allows you to configure the CAN interface.



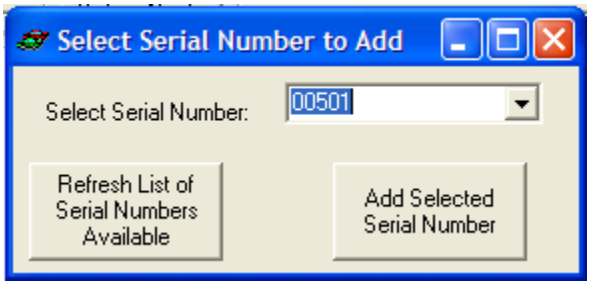
CONFIGURATION SCREEN
FIGURE 6.3.1

- **CAN Bit Rate:** Displays the CAN bit rate, from 100,000 bps to 1,000,000 bps. This is determined entirely by the bit timing parameters that are setup.

- **BS1, BS2, SJW, Prescaler:** These are the bit timing parameters that define the bit timing of the CAN messages. The sample point is BS1 / (BS1 + BS2 + 1). For example, if BS1=13 and BS2=1, then the sample point is 13/15 or 86.7% of the way through the bit. The SJW is the number of time quanta that the bit timing is allowed to vary. The Prescaler is used to determine the bit rate such that 30,000,000 / (Prescaler * (BS1 + BS2 + 1)) is equal to the CAN bus bit rate. When you change the values of BS1, BS2, or Prescaler, the settings are sent to the Model 9012, and the Model 9012 is queried for its bit rate and the response is reported on the screen as confirmation that the desired settings were arrived at.
- **CAN Flags:** The value that is entered in this field is interpreted in bit-wise fashion: bit 0 of this byte determines if automatic bus-off recovery is used. If this is 0, a bus off condition will persist. If this is 1, then a bus off condition can be cleared. Bit 1 of this byte indicates if the 120 ohm bus terminator is put in place. If this is 0, then no bus termination is used, such as if this device is in the middle of a chain of CAN devices. If this is a 1, then bus termination is used, which is appropriate in the case where the Model 9012 is at the end point of the physical CAN bus.
- **CAN Transmit Timeout:** This is the number of milliseconds that the 9012 will attempt to transmit a CAN message before giving up if it is not being acknowledged. If it gives up, the message XMIT TIMEOUT will appear after the transmitted message.
- **Stop on Any Error: If this mode is activated, when** the CAN/USB device detects any bus errors or communication errors, the device will automatically stop sending the repeating messages that it was previously commanded to send. This setting is stored in non-volatile storage on the CAN/USB device along with the mode of operation the CAN/USB device is in.
- **Ascending Count Test Sequence:** If this setting is used, then the 8th byte of the CAN payload will increment with each successive message. If the device is in receive mode, it will ensure that the 8th byte is steadily increasing, or a Receive Mismatch error will be displayed.

6.4. ADDING CAN NODES

It is necessary to specify the CAN nodes connected to the CAN bus. Use the ***Add a Node*** button on the main message screen to bring up the following screen:

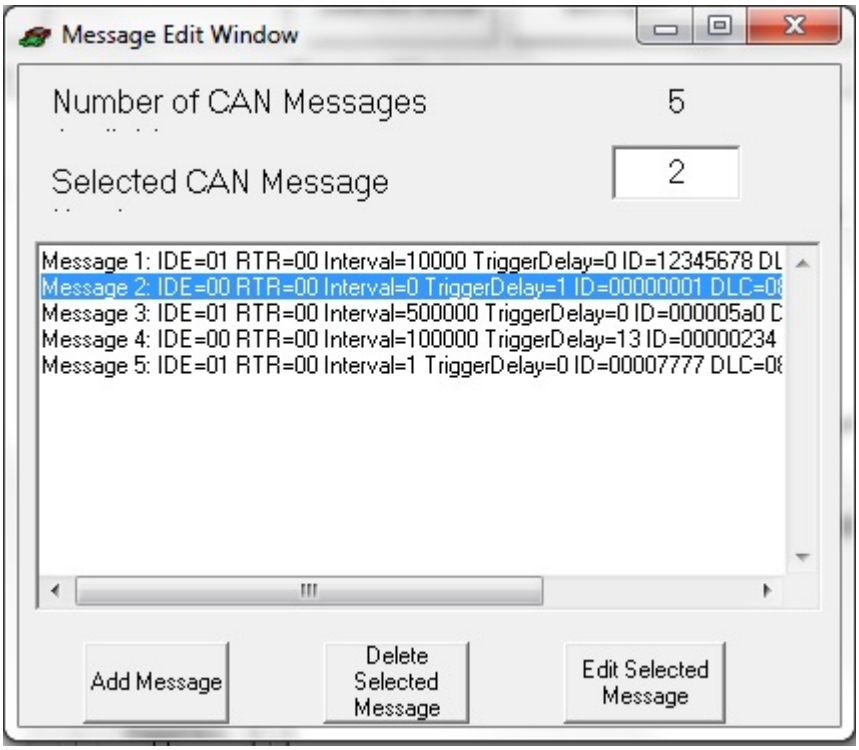


ADDING CAN NODES
FIGURE 6.4.1

To add a new serial number, click on the ***Add Selected Serial Number*** button. If another device has been connected since the screen was invoked, click on ***Refresh List of Serial Numbers Available*** to see if the new device is now present in the list. To see all the nodes that are identified as connected (or already appear in the node list) click on the drop-down box arrow to the right of the selection box and click on the node that you want to add. Note that this refresh button doesn't always work: it is sometimes necessary to re-open this ***Add Selected Serial Number*** box.

6.5. EDITING CAN MESSAGES

To view and edit messages within the CAN Message Center, press the ***Edit Messages*** button. The following screen will pop up:



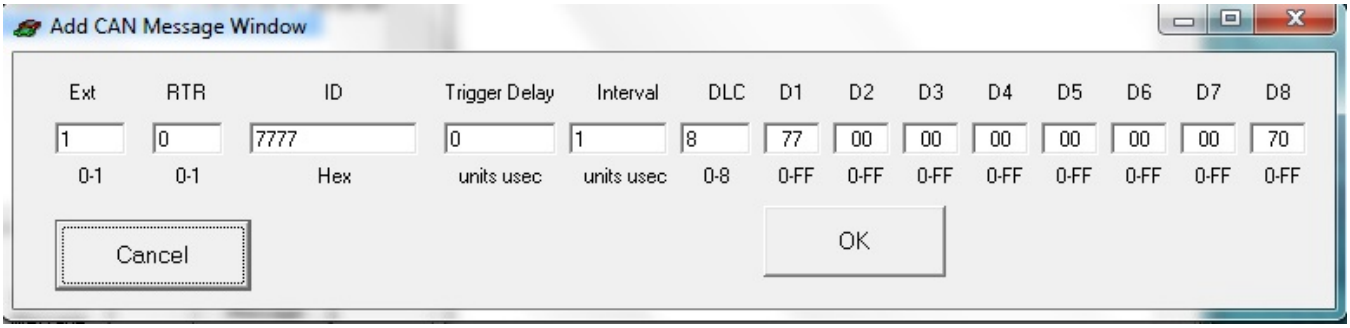
MESSAGE EDIT WINDOW
FIGURE 6.5.1.

To select a message, click on the message you would like to send next or to remove. That message will become highlighted (surrounded by gray). Now you can send the message by clicking on ***Send Selected Message*** on the main screen.

To remove a message, click on the message you want to remove, and press ***Delete Selected Message***.

6.6. ADDING A NEW CAN MESSAGE

To add a message, click on ***Add Message*** and the following screen will pop up:



ADD MESSAGE WINDOW
FIGURE 6.6.1.

In this screen you can enter a message that the CAN Message Center will be capable of sending later. Data entry is in hex.

When you are done entering the message, press the ***OK*** button and the message will appear in the ***Message Edit Window*** on the main message screen. The Ext field is either 0 or 1 to indicate a Standard Identifier (0) or an Extended Identifier (1). The RTR field is normally 0 but can bet set to 1 for a Remote frame. The ID contains the ID of the CAN message. The Trigger delay is 0 for no trigger or non-zero for a microsecond delay before the trigger pulse is sent before each transmission of this message. The Interval is either 0 for single occurring message or non-zero to indicate the number of microseconds that the converter box will attempt to place between each successive transmission of this message. A value of 1 indicates to send this message as fast as possible. DLC indicates the number of payload bytes that will be sent. The values of D1 through D8 indicate the payload bytes to be sent.

6.7. SENDING A CAN MESSAGE

To send the most recently selected message in the ***Message Edit Window***, press ***Send Selected Message*** on the main screen.

6.8. SENDING A CAN MESSAGE REPEATEDLY

You can send a message repeatedly by specifying a non-zero Interval in the message edit window. The message is sent at the interval specified in microseconds between each message. The device can be configured to stop sending any message if an error is encountered by checking the Stop on Error box or by pressing the Stop Sending Selected CAN Message button from the main screen.

7. TRIGGER OUTPUT SIGNAL

7.1. TRIG OUTPUT, PIN 8 ON CAN CONNECTOR

The Model 9012 CAN/USB Converter can be configured to output a pulse on its TRIGGER output when a specified CAN message event occurs, or upon command. This signal appears on pin 8 of the CAN connector.

7.2. TRIGGER SIGNAL DESCRIPTION

The idle state of the TRIGGER signal is 0 volts, referenced to CAN ground (pins 3 and 6 on the CAN connector).

The TRIGGER signal goes high to a level of approximately 4.6 to 5.0 volts when a programmed event occurs. It remains high for approximately 1.5 milliseconds, then returns to the 0-volt idle condition.

A connected oscilloscope or logic analyzer should be set up to trigger on the rising edge of the TRIGGER signal.

7.3. TRIGGER SIGNAL LOADING

The TRIGGER signal has a 10 KΩ resistor to CAN ground within the CAN/USB Converter. Pulses are generated by a current-limited high-side transistor powered by an internal isolated +5V supply within the unit. Maximum recommended load current is 1 milliamp. The high-side driver will go into current limit at approximately 4 milliamps.

8. BACKGROUND ON THE CAN BUS

8.1. OUTLINE SPECIFICATIONS

CAN (Controller Area Network) is typically implemented as a two-wire bus. In automotive applications, CAN typically runs at 500 kbps (SAE J2284). A single-wire, 33.3 kbps version (SAE J2411) is in use by GM but is not supported by the Model 9012.

In industrial applications, the CAN bus can run at up to 1 megabit per second.

CAN embodies a collision detection and arbitration mechanism that allows each node to start sending data whenever the bus is idle, and to resolve collisions without loss of data.

CAN has become increasingly dominant for medium-and high-speed automotive multiplexing, and is also widely used in industrial field bus networks.

8.2. RELATED PROTOCOLS

The SAE J1850 protocol was conceptually similar to CAN, in that it implemented collision detection and arbitration. However data encoding, bus speeds, and physical layer details differ from CAN. J1850-VPW (General Motors, Chrysler) involved a single-wire bus operating at an average data rate of 10,400 bps, using variable pulse-width data coding. J1850-PWM (Ford) implemented a two-wire bus operating at 41,600 bps, using pulse-width modulation. The American automotive manufacturers initially hoped that the costs of implementing J1850 would be significantly lower than the costs for CAN, but due in part to the failure of Ford, GM, and Chrysler to agree on compatible J1850 implementations, and in part to the increasing dominance of CAN, the American car manufacturers are tending to switch to CAN for new vehicle designs.

For lower-speed modules within a vehicle, the LIN bus protocol is becoming increasingly popular. CAN typically adds several dollars to the cost of an automotive ECU—including the cost of the CAN serial communications peripheral within the microcontroller; a CAN physical layer transceiver; and extra ROM and RAM to handle the higher layers of the protocol. In contrast, LIN is simpler, can use a UART peripheral as found on low-end microcontrollers, requires less hardware and software resources, but runs slower, 20 kilobits per second or less. LIN can be considered a sub-bus for handling body control functions, such as windows, doors, mirrors, and the like, while CAN covers high-speed functions such as communications between modules controlling the drive train. In an automotive vehicle, the body control network, running LIN, typically links up to the drive train network, running CAN, through a body control computer that acts as a bridge between LIN and CAN.

For further information on the LIN bus, please see information on our website regarding automotive data converters from Silicon Engines that support LIN.

8.3. PHYSICAL LAYER

At the two ends of a CAN network, the CAN bus should be terminated by a 120 ohm resistor, but in no other place. The Model 9012 CAN/USB Converter has the option of providing bus termination or not--fully under software control.

8.4. CAN CIRCUITRY WITHIN EACH ECU

A typical CAN ECU includes a microcontroller with a built-in CAN peripheral to handle the digital logic, and a CAN transceiver integrated circuit to handle analog signaling.

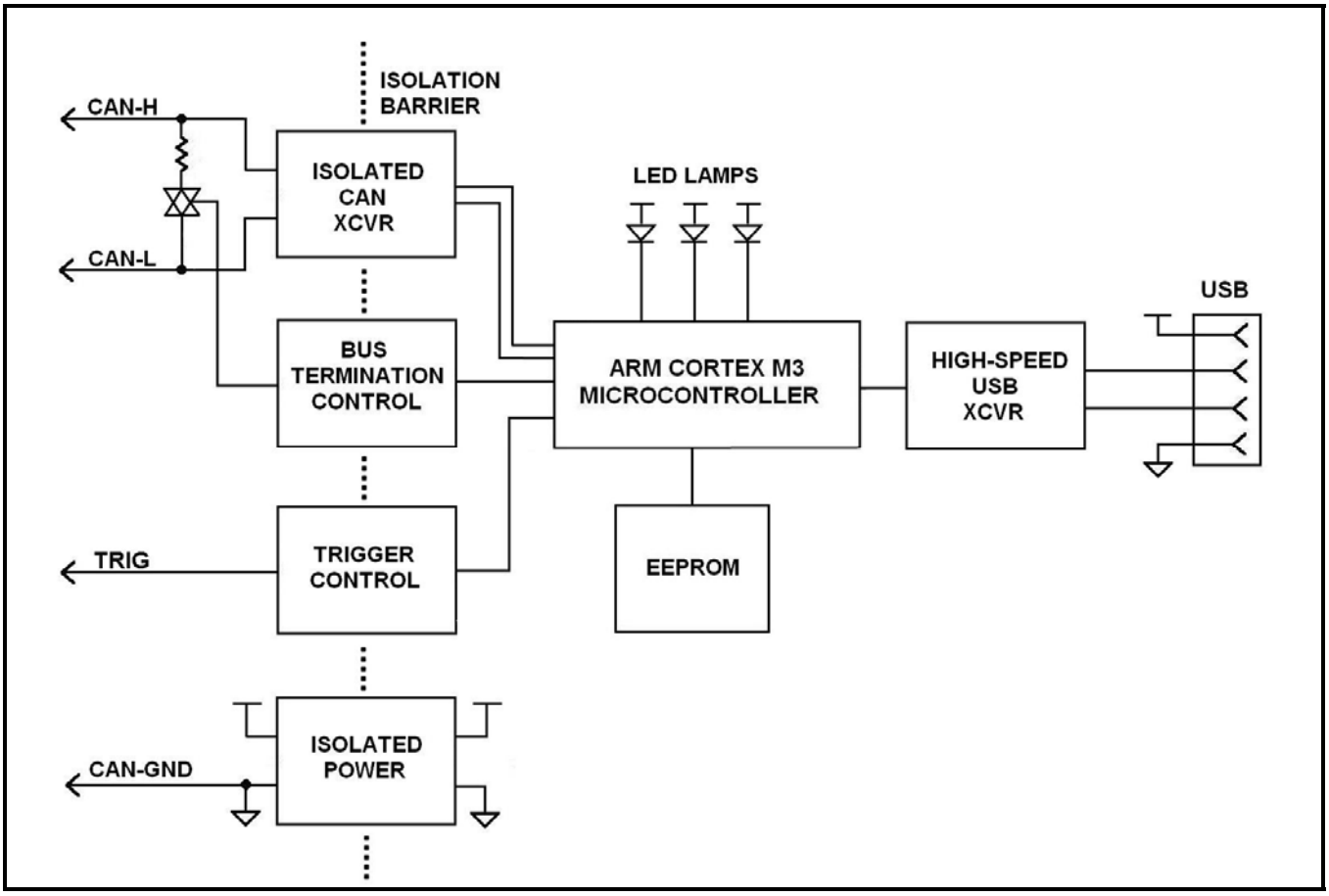
8.5. SERIAL DATA SPEED

Within a given CAN network, the CAN bus operates at a data rate between 100,000 to 1,000,000 bits per second (bps). The data speed for a given CAN network is fixed.

The Model CAN/USB Converter can be programmed for any data rate from 100,000 to 1,000,000 bps that can be reached by the programmed bit timing parameters. It should match the data rate selected for the target vehicle or industrial field bus.

9. MODEL 9012 HARDWARE

9.1. BLOCK DIAGRAM



MODEL 9012 BLOCK DIAGRAM
FIGURE 9.1.1.

9.2. HARDWARE DESCRIPTION

- **Microcontroller:** An STMicroelectronics STM32F205 microcontroller, a member of the 32-bit ARM Cortex M3 family running at 120 MHz, handles all the processing within the Model 9012.
- **EEPROM:** 8K by 8 serial EEPROM provides non-volatile storage for the CAN interface.
- **CAN interface:** Isolated CAN bus transceiver.
- **CAN bus termination:** 120 ohm bus termination resistor, switched on/off under software control.
- **High-speed USB:** High-speed USB 2.0 transceiver, operating at 480 mbps.

- **USB protection:** USB protective diode array to protect against static damage to the MCU.
- **Trigger output:** Can be set to trigger on any specific command identifier or upon command.

10. EMBEDDED SOFTWARE OUTLINE

10.1. CAN FEATURES

- **CAN bit rate:** CAN data speed programmable to any integer baud rate achieved by the bit timing parameters from 100,000 to 1,000,000 bps.
- **Frame length:** Automatically detects CAN messages of any length.
- **Frame errors:** Detects errors in the CAN frame.
- **Transmission errors:** Detects shorted CAN bus conditions—when the CAN message cannot be delivered.
- **EEPROM handler:** Stores configuration data in nonvolatile EEPROM memory.

10.2. FIRMWARE UPGRADES

The firmware inside the microcontroller is fully reprogrammable using the CAN/USB Message Center and the firmware files from the CD. Please view the README file installed with the program for details on the software upgrades available. The latest README file and software updates are also available on the Silicon Engines website.

11. REVISION HISTORY

11.1. REVISION A

Initial release.

☑