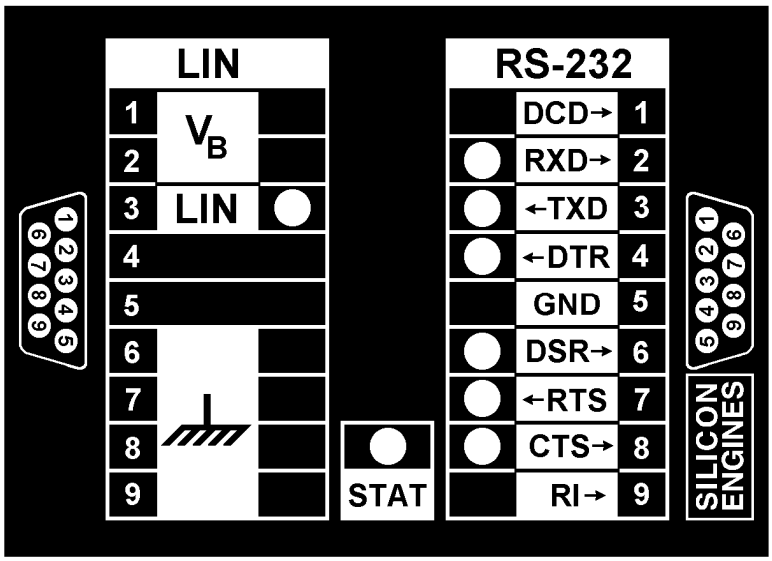
SILICON ENGINES

LIN/RS-232 CONVERTER, MODEL 9003

USER'S GUIDE



COMMENTS

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

SILICON ENGINES

3550 W. SALT CREEK LANE
SUITE 105
ARLINGTON HTS., IL 60005
847-637-1180
FAX 847-637-1185
sales@siliconengines.net
www.siliconengines.net

CONTENTS

1. OVERVIEW

1.1. INTRODUCTION

This document is the User's Guide for the Silicon Engines ***LIN/RS-232 Converter***, Model 9003, a compact electronic device that allows a personal computer to connect to an automotive communications data link compatible with the LIN protocol.

1.2. APPLICATIONS

- **Development:** Facilitates development of an automotive ECU (electronic control unit) that supports a LIN data bus, by allowing a personal computer to act as the LIN bus master during software development.
- **Production:** Allows the LIN data link to serve as a port for testing the ECU, and for downloading programs, parameters, serial number, calibration data, etc.
- **Service:** Allows a personal computer to act as a tester or diagnostic analyzer.

1.3. FUNCTIONS

- **Level conversion:** Converts signals between LIN voltage levels and RS-232 (CCITT V.24) levels, for connection to a personal computer.
- **LIN 2.0-2.1, LIN 1.3:** Supports the newer LIN 2.0 and 2.1 protocol versions as well as LIN 1.3.
- **Synchronization:** The LIN/RS-232 Converter generates the 13-bit synchronization break, synchronization field, and identifier field at the start of each LIN message frame. (*Note:* there is an alternate firmware version that creates a 19-bit break that can be used if that is required for compatibility with certain automotive ECUs; *see Sec. 7.2.*) A personal computer working alone cannot readily generate these signals because they depart from standard asynchronous data format and require precise synchronization timing.
- **Speed buffering:** Interfaces to the PC at a fixed baud rate of 38,400 bps. Interfaces to the ECU at a programmable LIN data rate, from 1,000 to 20,000 bps.
- **Automatic keep-alive:** Optionally keeps the LIN bus awake by generating periodic wakeup signals as defined by the LIN specifications. Optionally keeps the LIN bus awake indefinitely using an automatic wake-up mechanism. Also allows extended wakeups to be sent to ensure ECU's are kept awake.
- **Duplexing:** Converts the half-duplex LIN line, to standard full-duplex RS-232 signals. Avoids the need for the PC to deal with echoed characters.
- **Error codes:** Provides a number of error codes to help diagnose problems with LIN protocol issues.
- **Signal indicators:** Provides seven two-color LEDs to show the states of all significant signal lines.
- **Pass-through mode:** Provides an optional mode that allows the LIN RS232 converter to be used strictly as a level converter at 9600 baud for use with Freescale AN 2295 for programming a LIN module. A firmware variant also exists for performing pass-through mode at 19,200 baud.

The LIN/RS-232 Converter with a connected PC emulates a LIN master node, for connection to up to 15 LIN slave nodes. It cannot emulate a LIN slave node.

1.4. BACKGROUND ON LIN

LIN, ***Local Interconnect Network***, is a low-cost automotive multiplexing concept. Automotive electronic control modules (ECUs) that implement LIN communicate with each other over a one-wire data bus. In this respect, LIN is similar to other automotive multiplexing protocols—notably CAN and J1850. However LIN has been designed to realize lower ECU costs.

CAN (Controller Area Network) is typically implemented as a two-wire bus, running at 500 kbps (SAE J2284). A single-wire, 33.3 kbps version (SAE J2411) is in use by GM. 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 is becoming increasingly dominant for medium- and high-speed automotive multiplexing. However CAN adds several dollars to the cost of an automotive ECU—including the cost of the CAN serial communications controller within the microcontroller; a CAN physical layer transceiver; and extra ROM and RAM to handle the higher layers of the protocol.

The SAE J1850 protocol is conceptually similar to CAN, in that it implements collision detection and arbitration. However data encoding, bus speeds, and physical layer details differ from CAN. J1850-VPW (General Motors, Chrysler) involves a single-wire bus operating at an average data rate of 10,400 bps, using variable pulse-width data coding. J1850-PWM (Ford) implements a two-wire bus operating at 41,600 bps, using pulse-width modulation. The costs of implementing J1850 are similar to the costs for CAN, and the American car manufacturers are tending to switch to CAN for new vehicle designs.

There has been a growing perception of the need for a lower-cost network for many of the ECUs in a vehicle—for example, body control modules controlling doors, memory seats, and similar functions.

A consortium including Audi, BMW, DaimlerChrysler, Volkswagen, and Volvo, as well as Freescale Semiconductor (formerly Motorola) and Volcano Communications Technologies originally developed the LIN Protocol Specification.

LIN is a single-wire bus that can readily be implemented by very low-cost microcontrollers. The hardware required for a slave-mode LIN ECU is a standard SCI (serial communications interface) or UART (universal asynchronous receiver/transmitter). ECUs with low application processing requirements can even handle LIN without an SCI or UART. Data speed is 1,000 to 20,000 bps. The physical layer transceiver is very simple. There is a master node in a LIN network, keeping the protocol simple. Relatively little software overhead is required. In a typical vehicle architecture, the body control ECUs interconnect using LIN (rather than CAN or J1850). A master LIN module serves as the LIN bus master, and provides a bridge to the CAN network for interchange of information with higher-speed ECUs.

With respect to the physical layer, LIN is similar to ISO-9141, an automotive diagnostic standard published by the International Standards Organization in 1989. LIN is also similar to ISO-14230, an updated and extended diagnostic standard published in 1999, implementing ***Keyword Protocol 2000.*** However these diagnostic protocols are active only when a tester device is connected to the ECU, at the factory or in a service depot. LIN is a protocol for exchanging information between modules when the vehicle is moving down the road.

1.5. RELATED AUTOMOTIVE PROTOCOL CONVERTERS

In addition to the Model 9003 LIN/RS-232 Converter, Silicon Engines also offers:

- Model 9001, 9141/RS-232 Converter
- Model 9002, 14230/RS-232 Converter
- Model 9004, LIN/USB Converter
- Model 9009, 14230/USB Converter.

For details on these devices, check our web site, www.siliconengines.net, or send an email inquiry to sales@siliconengines.net.

2. LIN BUS INTRODUCTION

2.1. BUS TOPOLOGY

The LIN bus line is a bi-directional, half-duplex, serial input/output line for exchange of information between automotive ECUs (electronic control units).

Up to 16 modules may be connected on the LIN bus line. Recommended maximum wire length is 40 meters (131 feet) [*LIN Physical Layer Specification*, Rev. 2.1, Sec. 6.5.5, Table 6.11].

References to standards and specifications appear in this document in square brackets, such as: [LIN Protocol Specification]. Please see Part 13 for a list of references.

The LIN bus terminal on each ECU is connected to like terminals on other ECUs within the vehicle.

One LIN module acts as the master node. The other ECUs act as LIN slaves. Each LIN slave communicates over the LIN bus when it receives its unique ID code from the LIN master. There is no collision arbitration (unlike CAN or J1850).

2.2. PHYSICAL LAYER

Within the LIN bus master, the LIN bus is terminated to VBATT through a pull-up resistor with a value of 1 KΩ. Within each LIN slave, the LIN bus is terminated to VBATT by a pull-up resistor with a value of 30 KΩ. A diode is placed in series with each pull-up resistor, within each ECU, to prevent the LIN bus from providing a sneak path to power the ECU if it has lost its connection to VBATT [*LIN Physical Layer Specification*, Rev. 2.1, Sec. 6.5.1].

2.3. SERIAL DATA FORMAT

For the most part, the LIN bus uses asynchronous serial data format compatible with the SCI (serial communications interface) or UART (universal asynchronous receiver/transmitter) peripherals within popular microcontrollers. The data format is 8N1. Bits are sent in 10-bit words consisting of a START bit, eight data bits (least significant bit first), no parity bit, and one STOP bit [*LIN Physical Layer Specification*, Rev. 2.1, Sec. 6.4.3].

The data format is similar to RS-232, except that the serial data signals on the LIN bus are non-inverted, unipolar (VBATT and GROUND), while the serial data signals on the RS-232 TXD and RXD lines are inverted, bipolar (± 10 to 15 V).

BIT DESCRIPTION	LOGIC LEVEL	LIN VOLTAGE	RS-232 VOLTAGE
IDLE LINE	1	VBATT	-10 V
START BIT	0	GROUND	+10 V
DATA BITS	0	GROUND	+10 V
	1	VBATT	-10 V
STOP BIT	1	VBATT	-10 V

SERIAL DATA LEVELS FOR LIN, RS-232
FIGURE 2.3.1

2.4. LIN CIRCUITRY WITHIN EACH ECU

Within a typical LIN ECU, the LIN bus transmitter is an open-collector bipolar transistor (or open-drain MOSFET) that is normally off. The pull-up resistors to VBATT within the various LIN ECUs cause the LIN bus line to rise to the level of VBATT when the LIN bus is idle. An ECU selectively activates this transmitter transistor when transmitting on the LIN bus line.

The physical layer receiver within each ECU consists of a voltage comparator (or equivalent circuit) that slices at approximately 50% of VBATT to distinguish between high and low voltage levels on the LIN bus.

A number of semiconductor manufacturers offer LIN transceivers compatible with the above specifications. A list of LIN transceivers appears in the *REFERENCES* section of this document *(Part 13)*.

The LIN transmitter and receiver typically connect to an SCI (serial communications interface) or UART (universal asynchronous receiver-transmitter) within the ECU's microcontroller.

Many newer microcontrollers offer enhanced UARTs that can automatically handle the synchronization break feature that is special to LIN *(Sec. 2.6)*.

2.5. SERIAL DATA SPEED

Within a given vehicle, the LIN bus operates at a data rate from 1,000 to 20,000 bits per second (bps) [*LIN Physical Layer Specification*, Rev. 2.1, Sec. 6.3.] Three recommended LIN data rates were prescribed by an earlier version of the LIN specifications [*LIN Protocol Specification*, Rev. 1.3, *Bit rate*, p. 9].

SPEED	DATA RATE
SLOW	2400 BPS
MEDIUM	9600 BPS
FAST	19,200 BPS

LIN 1.3 RECOMMENDED STANDARD LIN BUS SPEEDS
FIGURE 2.5.1

The LIN/RS-232 Converter can be programmed for any data rate from 1,000 to 20,000 bps. It should match the data rate selected for the target vehicle.

2.6. HEADER FIELD

The LIN protocol requires a header field at the start of each message frame. The master LIN node always generates the header field. It consists of the following phases:

1. **Synchronization break:** A low-voltage break signal on the LIN bus lasting 13 nominal bit times at the effective data speed. This break symbol departs from the conventional asynchronous SCI/UART frame format.

2. **Synchronization break delimiter:** An idle (high-voltage) period on the LIN bus lasting one nominal bit time on the LIN bus.

3. **Synch field:** A single byte consisting of the hex value 55H, sent in standard 8N1 asynchronous data format, with one start bit, eight data bits, and a stop bit. This byte is sent at the LIN baud rate appropriate for the target ECUs. It is intended for use by slave ECUs that may not have accurate crystal-based clocks, allowing them to synchronize their responses with the LIN data rate established by the LIN bus master.

4. **Identifier field:** A single byte in 8N1 format that defines the content and length of the LIN message.

BIT NO.	BIT ID.	FUNCTION	COMMENTS
0	ID0	MESSAGE IDENTIFIER	LSB, SENT FIRST
1	ID1		
2	ID2		
3	ID3		
4	ID4		
5	ID5		
6	P0	PARITY BITS	
7	P1		MSB, SENT LAST

IDENTIFIER FIELD IN LIN MESSAGE HEADER
FIGURE 2.6.1

1. The least significant six bits, ID0 through ID5, specify a particular LIN message.

2. Bits P0 and P1 are parity check bits on the first six bits, ID0 through ID5. Each parity bit operates on four of the six bits ID0-ID5.

PARITY BIT	FORMULA	COMMENTS
P0	$ID1 \otimes ID1 \otimes ID2 \otimes ID4$	EVEN PARITY
P1	$\overline{ID1 \otimes ID3 \otimes ID4 \otimes ID5}$	ODD PARITY

PARITY CHECK BITS IN IDENTIFIER FIELD
FIGURE 2.6.2

The symbol ⊗ in the above chart denotes a logical exclusive OR operation.

2.7. LIN/RS-232 CONVERTER IS BUS MASTER

The LIN/RS-232 Converter, together with a connected personal computer, acts as the LIN bus master. It initiates each message frame, which can be completed by a slave device, or the LIN/RS-232 Converter itself as a slave task.

The LIN/RS-232 Converter with a connected PC emulates a LIN master node, for connection to up to 15 LIN slave nodes. It cannot emulate a LIN slave node.

2.8. HALF-DUPLEX SINGLE-WIRE INTERFACE

The LIN bus line is a single-wire interface. Data flow is half-duplex, in one direction only. Either the master ECU is talking, or a slave.

A direct result of the half-duplex nature of the LIN bus line is data echo. For example, whenever a LIN bus slave sends logic 0 to the LIN bus master on the LIN bus line, the LIN bus line goes low. The ECU's LIN bus line receiver detects this condition. This means that every byte that a LIN slave ECU sends to LIN master using its UART's TXD pin, is simultaneously echoed back to the ECU on its UART's RXD pin.

2.9. LIN BUS LINE LOGIC WITHIN LIN/RS-232 CONVERTER

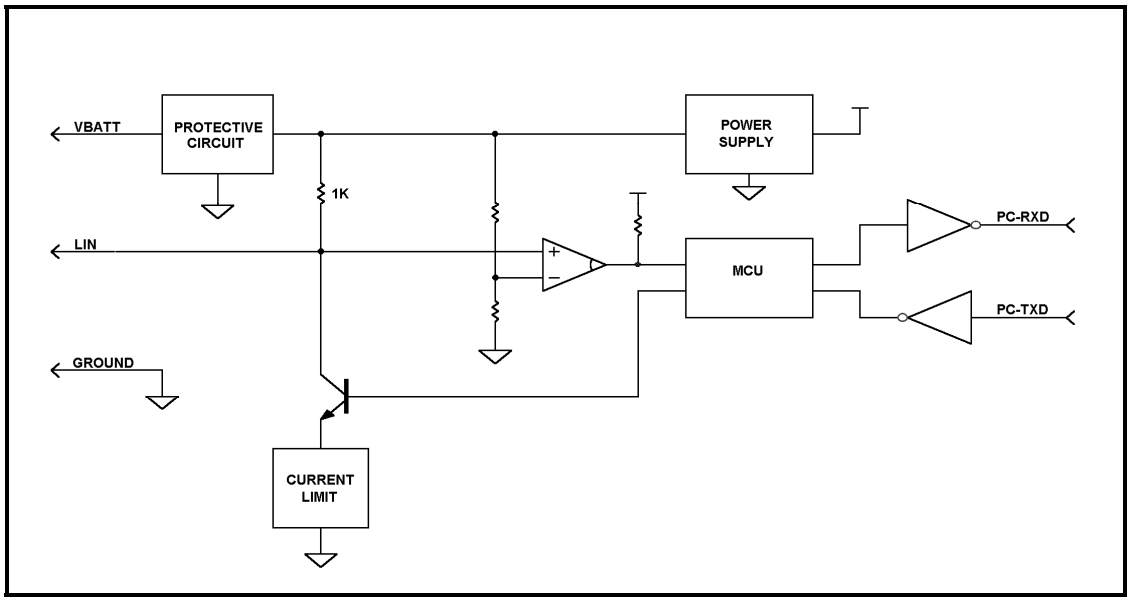
The LIN/RS-232 Converter acts as the LIN bus master in the system. A simplified block diagram for the LIN bus line logic within the LIN/RS-232 Converter appears below.

The microcontroller (MCU) within the Converter does baud-rate conversion. It sends and receives data over the RS-232 TXD and RXD lines at 38,400 bps, and sends and receives data over the LIN bus line at the programmable LIN baud rate of 1,000 to 20,000 bps.

The MCU also converts half-duplex LIN bus line data to full-duplex RS-232. Specifically, when the PC sends a command to transmit a message, to the LIN/RS-232 Converter, the MCU forwards it onto the LIN bus line. Because the LIN bus line is half-duplex, the LIN bus line comparator will see this transmitted data. The MCU filters out echoed bytes to help simplify PC software. The MCU also checks that the echoed bit is correct, and if not, it will generate an error code.

The LIN/RS-232 Converter contains protective circuitry to assure that the maximum voltage on the LIN bus line does not exceed 40 VDC, even in the presence of higher-voltage transients on the VBATT line.

The LIN/RS-232 Converter also contains current-limiting circuitry to protect its transmitter transistor against a short to VBATT. The maximum current is below the 200-mA limit specified in the standards [*LIN Physical Layer Specification,* Rev. 2.1, Sec. 6.5.4, Table 6.6, Parameter 12].



LIN BUS LINE LOGIC WITHIN LIN/RS-232 CONVERTER
FIGURE 2.9.1.

2.10. LIN BUS LINE ELECTRICAL SPECIFICATIONS

PARAMETER	MINIMUM	TYPICAL	MAXIMUM	CONDITIONS/COMMENTS
OPERATING VOLTAGE RANGE	+8 VDC	+13.8 VDC	+18 VDC	
LIN/RS-232 CONVERTER LOAD RESISTOR	950 Ω	1,000 Ω	1,050 Ω	1 KΩ ±5%
MAXIMUM VOLTAGE PLACED ON LIN BUS BY LIN/RS-232 CONVERTER		34 VDC		VOLTAGE CLAMP WITHIN LIN/RS-232 CONVERTER POWER SUPPLY
MAXIMUM SINK CURRENT, OUTPUT LOW		58 mA		CURRENT-LIMITED OUTPUT TRANSISTOR IN LIN/RS-232 CONVERTER

LIN/RS-232 CONVERTER ELECTRICAL SPECIFICATIONS
FIGURE 2.10.1

2.11. LIN BUS LINE CAPACITIVE LOADING

Bus line capacitance can be an important issue in LIN networks. The number of nodes, the cable capacitance, and the capacitive bus loads from the master and slave nodes all must be considered [*LIN Physical Layer Specification,* Revision 2.1, Sec. 6.5.5, Table 6.11].

PARAMETER	MINIMUM	TYPICAL	MAXIMUM	CONDITIONS/COMMENTS
TOTAL CAPACITANCE OF BUS	1	4	10	NANOFARADS
CAPACITANCE OF MASTER NODE		220		PICOFARADS
CAPACITANCE OF SLAVE NODES		220	250	PICOFARADS
LINE CAPACITANCE		100	150	PICOFARADS PER METER
LENGTH OF BUS LINE			40	METERS
NUMBER OF NODES			16	UNITS

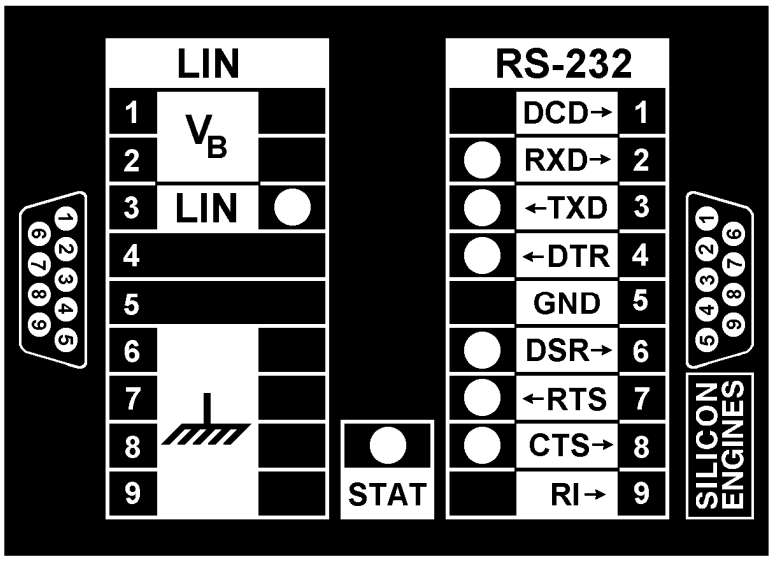
LIN BUS LINE CAPACITIVE LOADING
FIGURE 2.11.1

See Part 11 below for hints on troubleshooting problems with excessive capacitive bus loading.

3. ENCLOSURE

3.1. TOP PANEL

The LIN/RS-232 Converter is housed in a black plastic enclosure. Seven bicolor LED indicator lamps, and one green LED, appear through clear plastic windows on the top panel. The legends on the top panel identify the functions of these lamps, and identify the signals on the LIN and RS-232 connectors.



LIN/RS-232 CONVERTER TOP PANEL
FIGURE 3.1.1.

3.2. 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 3.2.1.

4. CONNECTORS

4.1. LIN CONNECTOR ON CONVERTER

The connector at the left of the LIN/RS-232 Converter is a type DB9M plug (9-pin male D sub-miniature).

A DB9F (female DB9) socket plugs in here. Three signals are supported: VBATT, the LIN bus line, and ground.

PIN NO.	SYMBOL	SIGNAL	DESCRIPTION
1-2	V_B	VBATT	BATTERY POWER
3	LIN	LIN BUS	LIN BUS LINE
4		NC	*NO CONNECTION*
5		NC	*NO CONNECTION*
6-9	⏚	GROUND	POWER AND SIGNAL RETURN

LIN CONNECTOR PIN-OUTS
FIGURE 4.1.1.

These pin-outs, as well as the locations of the pins within the 9-pin connector, are shown on the LIN/RS-232 Converter top panel legend *(Fig. 3.1.1).*

4.2. LIN CABLE TO ECU

Users must construct a special cable to connect from the ECU to the LIN/RS-232 Converter. Or contact Silicon Engines for assistance.

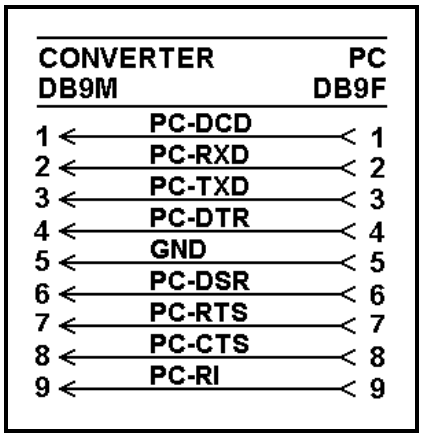
At the LIN/RS-232 Converter side, use a DB9F connector. Connect VBATT to pin 1, the LIN bus line to pin 3, and ground to pin 6. The locations of the pins within the connector are shown on the top panel decal *(see Fig. 3.1.1). (See Part 5 below for more information on VBATT and GND.)*

4.3. RS-232 CONNECTOR

The connector at the right of the LIN/RS-232 Converter is a DB9F type, designed for connection to an asynchronous serial communications port on a PC. Signal wires are shown on the top panel artwork *(Fig. 3.1.1).*

Connector polarities and signal names have been chosen for compatibility with the RS-232 (CCITT V.24) serial port of a recent model IBM®-compatible PC. PC-AT and later computer models typically provide a DB9M connector at their serial ports.

Typically the LIN/RS-232 Converter connects to the PC over a DB9 extender cable about 6 feet (2 M) long. A DB9M connector plugs into the LIN/RS-232 Converter at one end of the cable, and a DB9F connector plugs into the PC at the other end. This cable provides straight-through wiring—pin 1 connects to pin 1, pin 2 to pin 2, etc. A cable of this description is provided with each LIN/RS-232 Converter.



STRAIGHT-THROUGH RS-232 CABLE
FIGURE 4.3.1.

Notice: recommended practice is to connect the RI line on the LIN/RS-232 Converter to the RI line on the PC's RS-232 serial port. To prevent interference with the functioning of the LIN/RS-232 Converter, do not connect its RI signal in any other manner.

4.4. 25-PIN SERIAL PORT

If your computer provides a 25-pin DB25F RS-232 serial port connector, it will be necessary to install an adapter with a DB25M plug on one side, and a DB9M plug on the other, to adapt the computer to the LIN/RS-232 Converter. This type of adapter is commonly sold to adapt serial mouse devices to 25-pin ports, and should be readily available from the closest computer store—or contact Silicon Engines for assistance.

5. POWER REQUIREMENTS

5.1. CONNECTING VBATT AND GROUND

When working with an automotive ECU, the VBATT and GROUND lines are typically connected—along with the LIN bus line—to a suitable connector on the ECU. Both the LIN/RS-232 Converter and the ECU are powered from the same source of power.

The LIN/RS-232 Converter can also be powered by a DC power supply that connects to building AC power lines.

5.2. INPUT VOLTAGE AND CURRENT

The LIN/RS-232 Converter contains built-in power supply circuitry that generates needed power from VBATT and GND. The LIN/RS-232 Converter is compatible with 12-volt battery systems (8 to 18 VDC), as specified by [*LIN Physical Layer Specification*, Rev. 2.1, Sec. 6.5.4, Table 6.6, Parameter 9].

The unit generates +5 VDC internally for digital logic and the LED indicators, as well as ±10 VDC for the RS-232 interface.

SPECIFICATION	MIN.	TYP.	MAX	UNITS	CONDITIONS
SUPPLY VOLTAGE	8.0		18.0	VDC	CONTINUOUS OPERATION
			60	VDC	LOAD DUMP, 100 MS MAX.
			-60	VDC	REVERSE BATTERY
SUPPLY CURRENT		100		MA	VBATT=+12 VDC

SUPPLY POWER SPECIFICATIONS
FIGURE 5.2.1.

5.3. LOAD DUMP PROTECTION

The LIN/RS-232 Converter contains circuitry for protection against automotive load dump transients up to the maximum levels shown above. These levels are adequate for most current vehicle designs.

However, if higher transient levels are anticipated, measures should be taken to protect the LIN/RS-232 Converter. One method is to power the device from an AC line-powered DC power supply, rather than from the vehicle's battery.

5.4. SEPARATE VBATT SOURCES

If the ECU and the LIN/RS-232 Converter are powered from separate sources:

- **Grounds:** The ground of the LIN/RS-232 Converter must be connected to the grounds of the ECU and of both power sources.
- **VBATT:** The VBATT voltage provided to the LIN/RS-232 Converter should be within the ranges specified above, and within ±3 VDC of the VBATT voltage provided to the ECU.

5.5. REVERSE BATTERY PROTECTION

The LIN/RS-232 Converter is protected against inadvertent reverse battery connection. The unit will not operate properly with reversed power inputs, but will not be damaged, so long as the negative voltage is within the range specified above.

6. LAMP FUNCTIONS AND SIGNAL FLOW

6.1. STATUS LAMP

In the center of the top panel of the LIN/RS-232 Converter is a green LED lamp, marked **STAT**.

LED PATTERN	SIGNAL LINE CONDITION
STEADY GREEN	CONVERTER OPERATING NORMALLY
BLINKING	CONVERTER MCU ERROR

STATUS LAMP PATTERNS
FIGURE 6.1.1.

At power-up, the MCU within the LIN/RS-232 Converter does a quick self-diagnostic test. A steady STAT lamp indicates that the MCU has passed the test. If the lamp is blinking, please contact Silicon Engines for assistance.

6.2. TWO-COLOR LAMPS

Seven two-color indicators on the top panel of the LIN/RS-232 Converter show the status of key RS-232 and LIN signal lines. Each lamp glows either green or red whenever the device is powered up.

LED COLOR	SIGNAL LINE CONDITION
GREEN	HIGH VOLTAGE LEVEL ON SIGNAL AT CONNECTOR
RED	LOW VOLTAGE LEVEL ON SIGNAL AT CONNECTOR

LED COLOR CODES
FIGURE 6.2.1.

6.3. LIN BUS INDICATOR LAMP

The LIN bus indicator lamp indicates the voltage level on the LIN bus line.

CONN. PIN	SIGNAL NAME	DATA DIR.	LAMP COLOR	VOLTAGE LEVEL	SIGNAL FUNCTION
3	LIN	ECU↔PC	GREEN	VBATT	IDLE LINE; LOGIC 1 (MARK); STOP BIT
			RED	0 V	SYNCH BREAK; START BIT; LOGIC 0 (SPACE)

LIN BUS INDICATOR LAMP
FIGURE 6.3.1.

6.4. RS-232 SIGNAL LINES

Six LED indicators show the voltage levels on key RS-232 signal lines. The names of the RS-232 signals on the LIN/RS-232 Converter top panel are from the point of view of the connected personal computer.

CONN. PIN	SIGNAL NAME	DATA DIR.	LAMP COLOR	TYP. LEVEL	SIGNAL FUNCTION
1	DCD	CONV→PC	*(NONE)*	+10 V	DATA CARRIER DETECT, HIGH WHEN POWER IS ON
2	RXD	CONV→PC	GREEN	+10 V	START BIT; LOGIC 0 (SPACE)
			RED	-10 V	IDLE LINE; LOGIC 1 (MARK); STOP BIT
3	TXD	CONV←PC	GREEN	+10 V	START BIT; LOGIC 0 (SPACE)
			RED	-10 V	IDLE LINE; LOGIC 1 (MARK); STOP BIT
4	DTR	CONV←PC	GREEN	+10 V	*SIGNAL UNUSED BY LIN/RS-232 CONVERTER*
			RED	-10 V	*SIGNAL UNUSED BY LIN/RS-232 CONVERTER*
5	GND	---	*(NONE)*	0 V	SIGNAL COMMON
6	DSR	CONV→PC	GREEN	+10 V	LIN BUS AWAKE
			RED	-10 V	LIN BUS ASLEEP
7	RTS	CONV←PC	GREEN	+10 V	REQUEST TO SEND, FLOW CONTROL, PC READY TO RECEIVE FROM CONVERTER
			RED	-10 V	PC NOT READY TO RECEIVE FROM CONVERTER
8	CTS	CONV→PC	GREEN	+10 V	DATA SET READY, FLOW CONTROL, CONVERTER READY TO RECEIVE FROM PC
			RED	-10 V	CONVERTER NOT READY TO RECEIVE FROM PC
9	RI		(NONE)	0 V	RING INDICATOR, NOT USED FOR LIN COMMUNICATIONS , CONNECT ONLY TO PC-RI

RS-232 SIGNALS
FIGURE 6.4.1.

6.5. RXD: SERIAL DATA FROM CONVERTER TO PC

The RXD line leads to the RECEIVE DATA input of the PC's RS-232 serial port. Voltage levels are ±10 volts nominal. Data speed is fixed at 38,400 bps. Data format is 8N1—one start bit, eight data bits, no parity bit, and one stop bit.

6.6. TXD: SERIAL DATA FROM PC TO CONVERTER

The TXD line originates from the TRANSMIT DATA output of the PC's RS-232 serial port. Voltage levels are typically ±10 volts nominal. Data speed is fixed at 38,400 bps. Data format is 8N1—one start bit, eight data bits, no parity bit, and one stop bit.

6.7. DTR: UNUSED

The DTR (Data Terminal Ready) signal is not used by the LIN/RS-232 Converter. The PC may turn it on or off without any effect on the LIN/RS-232 Converter.

6.8. DSR: LIN BUS STATUS INDICATOR

The LIN/RS-232 Converter uses the DSR (DATA SET READY) line as a status signal to indicate whether the LIN bus is awake or sleep.

When DSR is high, the LIN bus is awake. Communications are active, or the bus is being kept alive by wake-up signals.

6.9. RTS: PC FLOW CONTROL

The PC can optionally use the RTS (REQUEST TO SEND) line for flow control. The PC sets RTS high to indicate that it is prepared to receive RS-232 data from the LIN/RS-232 Converter. The PC sets RTS low to cause the LIN/RS-232 Converter to pause in sending data to the PC.

The PC serial port should be configured for hardware flow control. RTS should be set ON except when desired to pause incoming data from the LIN/RS-232 Converter.

6.10. CTS: CONVERTER FLOW CONTROL

The LIN/RS-232 Converter uses the CTS (REQUEST TO SEND) line for flow control. The LIN/RS-232 Converter sets CTS high to indicate that it is prepared to receive RS-232 data from the PC. The LIN/RS-232 Converter sets CTS low to cause the PC to pause in sending data to the LIN/RS-232 Converter.

The PC serial port should be configured for hardware flow control. The LIN/RS-232 Converter will set CTS ON except when desired to pause incoming data from the PC.

6.11. RI: FACTORY TEST ONLY

The LIN/RS-232 Converter does not use the RI (RING INDICATOR) line during normal operations. This line is an input to the PC, typically used by an external modem to signal that there is an incoming call. The RI line is reserved for use by Silicon Engines for factory test purposes.

Notice: recommended practice is to connect the RI line on the LIN/RS-232 Converter to the RI line on the PC's RS-232 serial port. To prevent interference with the functioning of the LIN/RS-232 Converter, do not connect its RI signal in any other manner.

6.12. LAMP PATTERNS, SYSTEM IDLE

LIN LAMP				RS-232 LAMPS			
LAMP	**COLOR**	**STATE**	**LEVEL**	**LAMP**	**COLOR**	**STATE**	**LEVEL**
LIN	GREEN	IDLE	VBATT	RXD	RED	IDLE	-10 VDC
				TXD	RED	IDLE	-10 TO -15 VDC
				DTR	RED	OFF	-10 TO -15 VDC
				DSR	RED	OFF	-10 VDC
				RTS	RED	IDLE	-10 TO -15 VDC
				CTS	GREEN	READY	+10 VDC

LAMP PATTERNS, LIN SYSTEM IDLE
FIGURE 6.12.1.

The above lamp pattern should appear when the system is idle. The LIN/RS-232 Converter is connected to both the PC and the ECU, and the power is on. The LIN software in the PC is *not* running, so the RS-232 signal lines from the PC are all at idle levels. The RS-232 lamps are all red, while the LIN K lamp is green.

6.13. LIN LAMP STEADY RED

If the LIN lamp is red when the system is idle, check the ECU and the cables for a short to ground.

The LIN lamp should be green most of the time in any mode. When the system is communicating, the LIN lamp will be green while the LIN line is idle waiting for a new command, and during logic 1 data bits, stop bits, and inter-frame idle time. The LIN lamp should go red only for synchronization breaks, start bits, and logic 0 data bits. A steady red LIN lamp indicates a problem on the LIN bus line.

6.14. LAMP PATTERNS, LIN COMMUNICATIONS ACTIVE

LIN LAMP				RS-232 LAMPS			
LAMP	**COLOR**	**STATE**	**LEVEL**	**LAMP**	**COLOR**	**STATE**	**LEVEL**
LIN	GREEN WITH RED FLASHES	1/0	VBATT/ GND	RXD	RED WITH GREEN FLASHES	1/0	COMMUNICATING
				TXD	RED WITH GREEN FLASHES	1/0	COMMUNICATING
				DTR	RED OR GREEN	DON'T CARE	±10 TO 15 VDC
				DSR	GREEN	ON-- BUS AWAKE	+10 VDC
				RTS	GREEN	ON	+10 TO +15 VDC
				CTS	GREEN	ON	+10 VDC

LAMP PATTERNS, LIN COMMUNICATIONS ACTIVE
FIGURE 6.14.1.

7. FIRMWARE VARIANTS

7.1. PASS-THROUGH MODE AT 19,200 BAUD

Pass-Through Mode is a feature that allows the LIN/RS-232 Converter to be used strictly as a level translator to send data as-is onto the bus and present data from the bus as-is to the PC serial port. In a LIN/RS-232 Converter programmed with its standard firmware, when Pass-Through Mode is invoked, communications automatically take place at 9600 baud, in conformance with Freescale Application Note 2295 (*see Sec. 8.19*). To invoke Pass-Through Mode at 19,200 baud, the LIN/RS-232 Converter needs to be reprogrammed with a special firmware version that supports 19,200 baud Pass-Through Mode. The latest firmware versions available are listed in the file README.TXT that appears on the Silicon Engines CD in the UPGRADES folder. The instructions, utilities, and files necessary to perform this upgrade are also in the UPGRADES folder of the product CD.

7.2. 19-BIT BREAKS

In a LIN/RS-232 Converter programmed with standard firmware, the LIN/RS-232 Converter will send 13 bit breaks *(see Sec. 2.6).* This is in accordance with the LIN specifications. However, certain automotive modules require a longer break period. If you need a 19-bit break, then you should use a special firmware variant that supports 19-bit break. The latest firmware versions available are listed in the file README.TXT that appears on the Silicon Engines CD in the UPGRADES folder. The instructions, utilities, and files necessary to perform this upgrade are also in the UPGRADES folder of the product CD.

8. RS-232 PACKETS, PC TO CONVERTER

8.1. RS-232 COMMUNICATIONS

All communications with the PC take place over the RS-232 interface at 38,400 bps using 8N1 serial data format.

The PC sends messages in packets. The LIN/RS-232 Converter responds to each packet received with a response packet.

Note: As an exception, the LIN/RS-232 Converter can be put into Pass-Through Mode, communicating at 9600 bps without a packet structure *(see Sec. 8.19).*

8.2. HEADER, PC TO CONVERTER

Each packet sent by the PC to the Converter starts with a fixed five-byte header, as follows:

BYTE NO.	HEX VALUE	ASCII CHAR.
1	4CH	L
2	49H	I
3	4EH	N
4	43H	C
5	42H	B

INITIALIZATION COMMAND HEADER
FIGURE 8.2.1.

After the five-byte header, one or more data bytes are sent, with the specific values determined by the specific message contents, as described further below.

8.3. RS-232 IDENTIFIER BYTE

The sixth data byte, after the five-byte RS-232 header, is the RS-232 identifier byte.

If either of the two most significant bits of the RS-232 identifier byte is set, then this packet is an RS-232 command packet from the PC to the LIN/RS-232 Converter.

If both bit 7 and bit 6 are zero, then this packet is a LIN message packet that the LIN/RS-232 Converter will send out over the LIN bus using the currently effective LIN parameters. This message packet will not be processed as a Converter command.

BIT 7	BIT 6	MESSAGE TYPE
0	0	LIN MESSAGE PACKET
0	1	RS-232 COMMAND PACKET
1	0	
1	1	

BITS 7 AND 6 IN RS-232 IDENTIFIER BYTE
FIGURE 8.3.1

8.4. RS-232 COMMANDS

The following is a list of RS-232 commands.

COMMAND CODE	COMMAND DESCRIPTION
81H	CLEAR RECEIVE BUFFER
82H	SEND WAKE-UP SIGNAL
83H	QUERY STATUS
84H	QUERY REVISION
91H	SET CONTINUOUS WAKE-UP
92H	SET LIN BUS BAUD RATE
93H	SET WAKE-UP ENABLE
94H	SET PROTOCOL LEVEL
95H	SET EXTENDED WAKE-UP
96H	SET MASTER/SLAVE MODE
97H	SET FAST REPEAT INTERVAL
98H	SET FAST REPEAT MODE
E1H	SET PASS-THROUGH MODE

RS-232 COMMANDS
FIGURE 8.4.1

These are the only commands currently supported. If bit 7 or bit 6 is set in the RS-232 identifier byte, but the identifier does not specify one of the above commands, then the LIN/RS-232 Converter will return an error code.

8.5. LIN MESSAGE PACKET

If the RS-232 identifier byte does not have either bit 7 or bit 6 set, then it does not specify one of the above RS-232 commands from the PC to the LIN/RS-232 Converter. Rather this is a LIN message packet that will be forwarded directly onto the LIN bus. The Converter interprets the low-order six bits of the RS-232 identifier byte as the LIN message identifier. The LIN/RS-232 Converter calculates and inserts the P0 and P1 parity bits that the LIN ECU(s) will be expecting.

Following the RS-232 identifier byte, there is a two-byte length field, with the most significant byte first. If this length field is zero, then the LIN/RS-232 Converter will send out the identifier as a Master Task mode message. A LIN slave ECU is expected to complete this message.

BYTE NO.	HEX VALUE	COMMAND VALUE	COMMENTS
6	[ID]	00H ≤ ID ≤ 3FH	
7	MSB[N]	MOST SIG. BYTE, MSG. LENGTH	
8	LSB[N]	LEAST SIG. BYTE, MSG. LENGTH	
9	DATA	FIRST BYTE, MESSAGE DATA	MESSAGE DATA OPTIONAL
10	DATA	SECOND BYTE, MESSAGE DATA	MESSAGE DATA OPTIONAL
---	---	---	---
N+8	DATA	LAST BYTE, MESSAGE DATA	MESSAGE DATA OPTIONAL
N+9	Σ	CHECKSUM BYTE	
N+10	/Σ	INVERTED CHECKSUM BYTE	

LIN MESSAGE PACKET
FIGURE 8.5.1.

During a complete LIN transmission, the LIN/RS-232 Converter automatically generates the LIN checksum, and appends it to the message frame. Therefore the RS-232 packet from the PC should not include the LIN checksum.

8.6. CHECKSUM BYTES

At the end of each RS-232 packet are two checksum bytes. The first checksum byte is calculated as the sum of the preceding bytes in the command packet, including all the bytes in the packet, including the five-byte header, but excluding the two checksum bytes themselves. The addition is performed modulo-256—in other words, any overflow beyond 255 decimal is discarded.

The second checksum byte is the bitwise inversion of the first checksum byte. The two checksum bytes added together will equal FF hex.

In the command charts below, the symbol **Σ** indicates the checksum byte, and the symbol **/Σ** indicates the inverted checksum byte.

8.7. 81H: CLEAR CONVERTER RECEIVE BUFFER

This command clears any data that the LIN/RS-232 Converter has received over the LIN bus, but has not yet transmitted, because it was waiting for the RS-232 signal CTS (clear to send) to become active. There are no data bytes following this command. The LIN/RS-232 Converter will automatically send back a status code with acknowledgment after it receives this command.

BYTE NO.	HEX VALUE	COMMAND VALUE
6	81H	CLEAR RECEIVE BUFFER
7	Σ	CHECKSUM BYTE
8	/Σ	INVERTED CHECKSUM BYTE

CLEAR CONVERTER RECEIVE BUFFER COMMAND
FIGURE 8.7.1.

8.8. 82H: SEND WAKE-UP SIGNAL

This command wakes up the LIN bus. There are no data bytes following this command. The LIN/RS-232 Converter will automatically send back a status code with acknowledgment after it receives this command.

BYTE NO.	HEX VALUE	COMMAND VALUE
6	82H	SEND WAKE-UP SIGNAL
7	Σ	CHECKSUM BYTE
8	/Σ	INVERTED CHECKSUM BYTE

SEND WAKE-UP SIGNAL COMMAND
FIGURE 8.8.1.

Notes:

1. This command will be ignored by the LIN/RS-232 Converter if wake-ups are currently disabled through the 93H command *(Sec. 8.13).*

2. If the LIN bus was asleep when this command is received, and wake-up signals are currently enabled (93H command, *Sec. 8.13*), the LIN/RS-232 Converter will automatically send a wake-up signal when necessary to wake up the LIN bus. PC application software could utilize this 82H command as a keep-awake signal when there is no other LIN message traffic.

8.9. 83H: QUERY STATUS

This command causes the LIN/RS-232 Converter to send back a status code with acknowledgment. There are no data bytes following this command.

BYTE NO.	HEX VALUE	COMMAND VALUE
6	83H	QUERY STATUS
7	Σ	CHECKSUM BYTE
8	/Σ	INVERTED CHECKSUM BYTE

QUERY STATUS COMMAND
FIGURE 8.9.1.

8.10. 84H: QUERY REVISION

This command causes the LIN/RS-232 Converter to send back a revision code with a special header. The purpose is to notify the user as to the current LIN/RS-232 Converter firmware and hardware revision. New revisions of the firmware are made from time to time to add features or correct issues found in earlier revisions. There are no data bytes following the 84H command. Note that the status code does not change when this command is issued.

BYTE NO.	HEX VALUE	COMMAND VALUE
6	84H	QUERY REVISION
7	Σ	CHECKSUM BYTE
8	/Σ	INVERTED CHECKSUM BYTE

QUERY REVISION COMMAND
FIGURE 8.10.1.

8.11. 91H: SET CONTINUOUS WAKE-UPS

This command controls whether the LIN/RS-232 Converter will generate automatic wake-up messages on a continuous basis. This command has a single byte data field. If the data byte is 01 hex, then continuous wake-ups are enabled. If the data byte is zero, continuous wake-ups are disabled.

BYTE NO.	HEX VALUE	COMMAND VALUE
6	91H	SET CONTINUOUS WAKE-UPS
7	00H OR 01H	00H = CONTINUOUS WAKE-UPS OFF; 01H = CONTINUOUS WAKE-UPS ON
8	Σ	CHECKSUM BYTE
9	/Σ	INVERTED CHECKSUM BYTE

COMMAND TO ENABLE OR DISABLE CONTINUOUS WAKE-UP SIGNALS
FIGURE 8.11.1.

The LIN/RS-232 Converter will automatically send back a status code with acknowledgment after it receives this command.

Notes:

1. This command will be ignored by the LIN/RS-232 Converter if wake-ups are currently disabled through the 93H command *(Sec. 8.13).*

2. When continuous wake-ups are enabled, the LIN/RS-232 Converter will automatically send a new wake-up signal over the LIN bus whenever it detects that the bus has been idle for a specified period. (If configured for LIN 1.3, 128 bit times at the effective LIN bus data rate; if configured for LIN 2.x, 150 milliseconds.) If no LIN bus traffic is detected in response, the LIN/RS-232 Converter tries again. After three tries of sending a wake-up signal, the LIN/RS-232 Converter waits a specified period (LIN 1.3: 15,000 bit times; LIN 2.x: 1500 milliseconds), and then rechecks to see whether continuous wake-ups are still enabled. If continuous wake-ups are enabled by the 91H command, then the Converter automatically retries the wake-up sequence. If continuous wake-ups are disabled, then the Converter goes to sleep.

3. The default setting is zero—continuous wake-ups are off.

8.12. 92H: SET LIN BUS BAUD RATE

This command sets the data communications rate for the LIN bus. Valid baud rates are between 1,000 and 20,000 bits per second.

There are two data bytes following this command. The first data byte is the most significant byte of the LIN baud rate. The second data byte is the least significant byte of the LIN baud rate. After this command is sent to the LIN/RS-232 Converter, it will send back a status code and acknowledgment.

BYTE NO.	HEX VALUE	COMMAND VALUE
6	92H	SET LIN BUS BAUD RATE
7	25H*	MOST SIG. BYTE, BAUD RATE
8	80H*	LEAST SIG. BYTE, BAUD RATE
9	Σ	CHECKSUM BYTE
10	/Σ	INVERTED CHECKSUM BYTE
* EXAMPLE VALUES FOR 9600 BAUD LIN DATA RATE		

SET LIN BUS BAUD RATE
FIGURE 8.12.1.

8.13. 93H: SET WAKE-UP ENABLE

This command is supported by LIN/RS-232 Converter firmware Rev. 9 or later, and PC software revision 1.6 or later. Please contact Silicon Engines if you need assistance in updating older versions.

This command controls whether the LIN/RS-232 Converter will generate wake-up messages.

BYTE NO.	HEX VALUE	COMMAND VALUE
6	93H	ENABLE WAKE-UPS
7	00H OR 01H	00H = WAKE-UPS DISABLED, 01H = WAKE-UPS ENABLED
8	Σ	CHECKSUM BYTE
9	/Σ	INVERTED CHECKSUM BYTE

COMMAND TO ENABLE OR DISABLE WAKE-UP SIGNALS
FIGURE 8.13.1.

The LIN/RS-232 Converter will automatically send back a status code with acknowledgment after it receives this command.

Notes:

1. This 93H command enables or disables the wake-up features of the individual 82H wake-up command *(Sec. 8.8)* and of the continuous 91H wake-up command *(Sec. 8.11).*

2. The default setting is zero, wake-ups are disabled.

8.14. 94H: SET LIN PROTOCOL LEVEL

This command is supported by LIN/RS-232 Converter firmware Rev. 14 or later, and PC software revision 2.0 or later. Please contact Silicon Engines if you need assistance in updating older versions.

This command controls whether the LIN/RS-232 Converter operates according to LIN Protocol 2.0-2.1, or according to version 1.3.

BYTE NO.	HEX VALUE	COMMAND VALUE
6	94H	SET LIN PROTOCOL LEVEL
7	00H OR 01H	00H = LIN 1.3, 01H = LIN 2.0 - 2.1
8	Σ	CHECKSUM BYTE
9	/Σ	INVERTED CHECKSUM BYTE

COMMAND TO SET PROTOCOL LEVEL
FIGURE 8.14.1.

The LIN/RS-232 Converter will automatically send back a status code with acknowledgment after it receives this command.

In LIN 1.3 mode, the LIN/RS-232 Converter generates a *classic checksum* at the end of each packet. It covers just the data bytes in the packet.

In LIN 2.0 – 2.1 mode, the LIN/RS-232 Converter generates an *enhanced checksum* at the end of each packet. It covers all the data bytes in the packet, as well as the packet identifier. As an exception, however, even when set for LIN 2.0 - 2.1 mode, the classic checksum is used when the packet identifier is in the range 60 to 63 (3C through 3F hex). [*Lin Protocol Specification,* Rev. 2.1, Sec. 2.3.2.5.]

Also, in LIN 2.0 - 2.1 mode, different time periods apply for wake-up transmissions *(Sec. 8.11)*.

8.15. 95H: SET EXTENDED WAKE-UPS

This command is supported by LIN/RS-232 Converter firmware Rev. 28 or later, and PC software revision 2.0.9 or later. Please contact Silicon Engines if you need assistance in updating older versions.

This command controls whether the LIN/RS-232 Converter uses standard wake-up signals, as defined by the LIN specification, or extended wake-up signals. Extended wake-up signals include the standard wake-up signal, as defined by the LIN specification, as well as a synchronization break symbol and a synchronization header (55H) as well. This is intended to be used with ECUs that require more than just the regular wake-up signal to keep awake.

BYTE NO.	HEX VALUE	COMMAND VALUE
6	95H	SET EXTENDED WAKE-UPS
7	00H OR 01H	00H = DISABLED 01H = ENABLED
8	Σ	CHECKSUM BYTE
9	/Σ	INVERTED CHECKSUM BYTE

COMMAND TO SET EXTENDED WAKE-UPS
FIGURE 8.15.1.

8.16. 96H: SET MASTER/SLAVE MODE

This command is supported by LIN/RS-232 Converter firmware Rev. 32 or later, and PC software revision 2.1.0 or later. Please contact Silicon Engines if you need assistance in updating older versions.

This command controls whether the LIN/RS-232 Converter is the Master node on the bus or whether it is a Slave or Listener device on the bus. It controls which pull-up resistor to use on the bus. The default is Master configuration, which was the only option for LIN/RS-232 Converters before firmware Revision 32. Now you can set the device to operate in Slave/Listener configuration, in order to receive data when there is a different Master node on the bus.

BYTE NO.	HEX VALUE	COMMAND VALUE
6	96H	SET MASTER/SLAVE MODE
7	00H OR 01H	00H = SLAVE/LISTENER 01H = MASTER
8	Σ	CHECKSUM BYTE
9	/Σ	INVERTED CHECKSUM BYTE

COMMAND TO SET MASTER/SLAVE MODE
FIGURE 8.16.1.

8.17. 97H: SET FAST REPEAT INTERVAL

This command is supported by LIN/RS-232 Converter firmware Rev. 34 or later, and PC software revision 2.1.1 or later. Please contact Silicon Engines if you need assistance in updating older versions.

This command sets the time interval between transmissions of a Fast Repeat message. It is in units of milliseconds and can be set in the range from 0 to 5000. A Fast Repeat message is a message that has been prefaced with a Fast Repeat command which will cause the transmitted message to be repeated forever until a Clear Buffers command is sent. The default value for this parameter is 0.

BYTE NO.	HEX VALUE	COMMAND VALUE
6	97H	SET FAST REPEAT INTERVAL
7	X	MSB OF FAST REPEAT INTERVAL
8	X	LSB OF FAST REPEAT INTERVAL
9	Σ	CHECKSUM BYTE
10	/Σ	INVERTED CHECKSUM BYTE

COMMAND TO SET FAST REPEAT INTERVAL
FIGURE 8.17.1.

8.18. 98H: SET FAST REPEAT MODE

This command is supported by LIN/RS-232 Converter firmware Rev. 34 or later, and PC software revision 2.1.1 or later. Please contact Silicon Engines if you need assistance in updating older versions.

This command sets the LIN/RS-232 Converter in Fast Repeat mode. The next transmitted message will be repeatedly transmitted at the time interval specified by the Set Fast Repeat Interval command. The Converter will stop sending this message only when a Clear Buffers command is sent. While the message is being sent, acknowledgements of each transmitted message will NOT be returned to the PC, BUT if the message being transmitted is just a master task byte with a slave device filling in the rest of the data, then the data IS returned to the PC. While in Fast Repeat mode, you cannot transmit any more messages or configure the Converter with configuration commands until a Clear Buffers command is sent.

BYTE NO.	HEX VALUE	COMMAND VALUE
6	98H	SET FAST REPEAT MODE
7	Σ	CHECKSUM BYTE
8	/Σ	INVERTED CHECKSUM BYTE

COMMAND TO SET FAST REPEAT MODE
FIGURE 8.18.1.

8.19. E1H: INVOKE PASS-THROUGH MODE

This command is supported by LIN/RS-232 Converter firmware Rev. 27 or later. This mode is compatible with programs LINPASSVB, PASSTEST, and PASSTEST2 which are available on the CD that is shipped with this product. Please contact Silicon Engines if you need assistance in updating older versions.

This command forces the LIN/RS-232 Converter into a special mode that is not aborted until the box is physically reset by removing and re-applying power. This mode is called Pass-Through Mode. Communications with the LIN/RS232 Converter, after this command is sent, switch to 9600 bps. All bytes and break symbols transmitted on the RS-232 transmit line from the PC at 9600 bps will be put on the LIN bus at 9600 bps, and, all traffic on the LIN bus at 9600 bps, including breaks, will be transmitted out to the PC's RS232 receive line at 9600 bps. Breaks are transmitted on the LIN bus as ten 0 bits. This mode is compatible with Freescale Application Note 2295 for use in programming Freescale LIN nodes using software provided by Freescale. In order to exercise the mode, first run a program such as LINPASSVB (available on the CD that is shipped with this product), and then run the program from Freescale (or any similar custom program).

Note that the command to invoke Pass-Through Mode has two bytes as arguments that are used to unlock this command. The command must be entered precisely as shown in the figure below to invoke Pass Through Mode. There is no response to the Pass-Through Mode command, since the Converter has fundamentally changed into a mode without a built-in packetized structure.

BYTE NO.	HEX VALUE	COMMAND VALUE
6	E1H	SET LIN PROTOCOL LEVEL
7	0EH	UNLOCK BYTE 1 = 0EH
8	B5H	UNLOCK BYTE 2 = B5H
9	Σ	CHECKSUM BYTE
10	/Σ	INVERTED CHECKSUM BYTE

COMMAND TO SET PROTOCOL LEVEL
FIGURE 8.19.1.

9. RS-232 PACKETS, CONVERTER TO PC

9.1. RESPONSE PACKET FORMAT

The LIN/RS-232 Converter sends response packets to the PC, over the RS-232 interface, at 38,400 bps, only in response to RS-232 packets received from the PC.

Each response packet starts with a fixed five-byte header, as follows:

BYTE NO.	HEX VALUE	ASCII CHAR.
1	52H	R
2	45H	E
3	53H	S
4	50H	P
5	3AH	:

RESPONSE PACKET HEADER
FIGURE 9.1.1.

One or more data bytes may be included in the response packet, depending on the type of packet involved.

The response packet is terminated by two checksum bytes, calculated in the same way as for RS-232 command packets, as described above.

9.2. RS-232 IDENTIFIER BYTE

The sixth data byte, after the five-byte RS-232 header, is the RS-232 identifier byte.

If either of the two most significant bits of the RS-232 identifier byte is set, then this packet is a response packet, sent by the LIN/RS-232 Converter as a response to a previously received RS-232 command packet from the PC.

If both bit 7 and bit 6 are zero, then this packet is a LIN message packet that the LIN/RS-232 Converter has received over the LIN bus from an ECU.

BIT 7	BIT 6	MESSAGE TYPE
0	0	LIN MESSAGE PACKET
0	1	RS-232 RESPONSE PACKET
1	0	
1	1	

BITS 7 AND 6 IN RS-232 IDENTIFIER BYTE
FIGURE 9.2.1

9.3. RS-232 RESPONSE PACKETS

The following is a list of RS-232 response packets:

RESPONSE CODE	RESPONSE DESCRIPTION
81H	REVISION LEVEL RESPONSE
82H	STATUS CODE RESPONSE

RS-232 RESPONSE CODES
FIGURE 9.3.1

9.4. LIN MESSAGE PACKET

If the RS-232 identifier byte does not have either bit 7 or bit 6 set, then this is a LIN message packet that the Converter has received from a slave ECU.

The message data is embodied in the LIN message packet as follows. The first four bytes in the data field of the packet are data bytes 1 through 4. Four bytes appear in the packet even if there were less than four data bytes in the received LIN message. After every four data bytes, there is a Length Indicator byte that lets the PC know how many of the previous bytes were data bytes, and whether to continue receiving bytes in the data field. The Length Indicator can be 00H to 04H to show the number of valid data bytes in the previous four; or can be 88H to indicate that a frame error has been detected, or can be FFH to indicate a continuing message.

In order to provide an economical LIN/RS-232 Converter, we have used a low-cost microcontroller, so the message buffering capability within LIN/RS-232 Converter is limited to only 16 bytes. The PC has a huge buffering capability that cannot cost-effectively be achieved inside the LIN Converter. Therefore we rely on the PC to provide storage for long messages.

The LIN/RS-232 Converter will immediately begin sending a message to the PC after it receives the first data byte from a slave task directly following an ID sent by a master task. Since the Converter does not know beforehand the length of the message, it sends the bytes to the PC as quickly as it can. This includes the LIN checksum byte. The Converter makes no attempt to check the LIN checksum byte. The PC host program should perform this task.

The LIN/RS-232 Converter will terminate a message if it receives a new synchronization pattern on the LIN bus, or if it detects a timeout of 64 bit times.

Here is an example. Assume that the following message appears on the LIN bus:

LIN BUS DATA (HEX)
SYNCH, C1, 11, 22, 33, 44, 55, 66, 99

EXAMPLE LIN BUS BYTES
FIGURE 9.4.1.

The LIN/RS-232 Converter sends the following packet to the PC:

BYTE NO.	HEX VALUE	ASCII CHAR.	BYTE DESCRIPTION
1	52	R	HEADER BYTE #1
2	45	E	HEADER BYTE #2
3	53	S	HEADER BYTE #3
4	50	P	HEADER BYTE #4
5	3A	:	HEADER BYTE #5
6	01		ID/RESPONSE TYPE
7	11		MESSAGE DATA BYTE #1
8	22		MESSAGE DATA BYTE #2
9	33		MESSAGE DATA BYTE #3
10	44		MESSAGE DATA BYTE #4
11	FF		LENGTH/CONTINUATION INDICATOR
12	55		MESSAGE DATA BYTE #5
13	66		MESSAGE DATA BYTE #6
14	99		MESSAGE DATA BYTE #7 (CHECKSUM)
15	00		MESSAGE DATA BYTE #8 (UNUSED)
16	03		LENGTH/CONTINUATION INDICATOR
17	75		CHECKSUM BYTE
18	8A		INVERTED CHECKSUM BYTE

EXAMPLE LIN MESSAGE PACKET
FIGURE 9.4.2.

As long as the PC asserts the RS-232 signal CTS to the Converter, it will continue sending all LIN data. If the LIN/RS-232 Converter overflows its 16-byte buffer while CTS is de-asserted, an error condition will result.

9.5. 81H: REVISION LEVEL RESPONSE PACKET

The LIN/RS-232 Converter sends this response packet in answer to an 84H Query Revision command from the PC.

BYTE NO.	HEX VALUE	COMMAND VALUE
6	81H	REVISION LEVEL RESPONSE
7	[REV.]	REVISION LEVEL CODE
8	Σ	CHECKSUM BYTE
9	/Σ	INVERTED CHECKSUM BYTE

REVISION LEVEL RESPONSE PACKET
FIGURE 9.5.1.

9.6. 82H: STATUS CODE RESPONSE PACKET

The LIN/RS-232 Converter sends this response packet in answer to a variety of commands received from the PC.

BYTE NO.	HEX VALUE	COMMAND VALUE
6	82H	STATUS CODE RESPONSE
7	[STATUS CODE]	STATUS CODE (SEE NEXT CHART BELOW)
8	[AWAKE CODE]	00H = LIN BUS ASLEEP 01H = LIN BUS AWAKE
9	[COMMAND CODE]	COMMAND CODE BEING RESPONDED TO
10	Σ	CHECKSUM BYTE
11	/Σ	INVERTED CHECKSUM BYTE

STATUS CODE RESPONSE PACKET
FIGURE 9.6.1.

The following is a list of status codes that can appear in a status code response packet:

HEX VALUE	COMMAND MEANING	LIKELY DIAGNOSIS/ COMMENTS
00H	CONVERTER FAILED TO BRING LIN BUS LINE LOW	LIN BUS LINE SHORTED TO VBATT
01H	IDENTIFIER PARITY ERROR DURING RECEPTION	COMMUNICATIONS NOISE, BUS CONTENTION, ECU FIRMWARE ERROR
02H	INCONSISTENT SYNCH FIELD DURING RECEPTION	COMMUNICATIONS NOISE, BUS CONTENTION, ECU FIRMWARE ERROR
03H	FRAME ERROR DURING RECEPTION	STOP BIT AT 9-11 BITS
04H	LIN BUS LINE FAILED TO RETURN HIGH	BUS CONTENTION
05H	RECEIVED ILLEGAL DATA WHILE BUS WAS ASLEEP	ECU SOFTWARE ERROR
06H	SLAVE NOT RESPONDING	64 BIT TIMES ALLOWED. ECU ERROR, WIRING ERROR.
20H	RECEIVED INVALID CHECKSUM FROM PC	COMMUNICATIONS NOISE, PC SOFTWARE ERROR
21H	RECEIVED INVALID COMMAND FROM PC	COMMUNICATIONS NOISE, PC SOFTWARE ERROR
22H	COMMAND NOT ACCEPTABLE IN FAST REPEAT MODE	MODE ERROR (USE CLEAR BUFFERS COMMAND TO EXIT FAST REPEAT MODE)
30H	RECEIVE BUFFER OVERFLOW (FROM PC-TXD TO LIN BOX)	FLOW CONTROL FAILURE
31H	TRANSMIT BUFFER OVERFLOW (FROM LIN BOX TO PC-RXD)	FLOW CONTROL FAILURE
4FH	NO ERROR PRESENT	SYSTEM OPERATING PROPERLY
50H	LIN BUS TRANSMIT BUFFER OVERFLOW (FROM PC TO LIN BUS)	FLOW CONTROL ERROR
51H	LIN BUS RECEIVE BUFFER OVERFLOW (FROM LIN BUS TO PC)	FLOW CONTROL ERROR
E0H	CONVERTER MCU ROM CHECKSUM ERROR	HARDWARE FAULT, CONTACT SILICON ENGINES

LIN/RS-232 CONVERTER RESPONSE CODES
FIGURE 9.6.2.

10. LIN MESSAGE CENTER

10.1. WINDOWS SOFTWARE

The ***LIN Message Center*** is a Microsoft® Windows® utility for communicating with the LIN/RS-232 Converter. A copy of this software is provided by Silicon Engines on a compact disc with each LIN/RS-232 Converter shipment. This program is compatible with Windows 95, 98, ME, XP, and Vista.

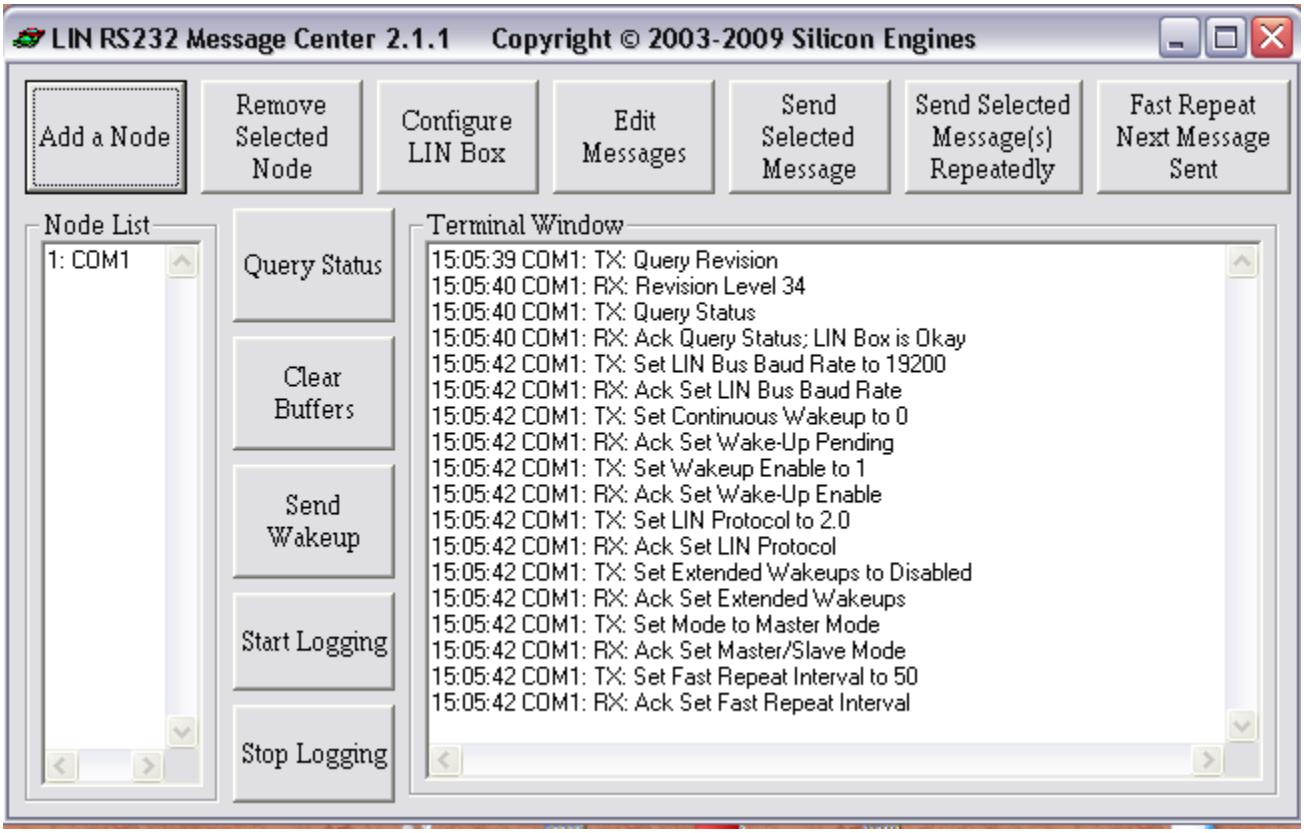
Windows Vista is supported by LIN Message Center revision 2.0.7 or later. Please contact Silicon Engines if you need assistance in updating older versions.

10.2. INSTALLATION

Insert the CD into your computer's CD drive. Run SETUP.EXE from the CD and follow the instructions provided.

10.3. LIN MESSAGE CENTER SCREEN

When you first start the LIN Message Center program, it will pop up a screen with this general format:



EXAMPLE LIN MESSAGE CENTER SCREEN
FIGURE 10.3.1.

10.4. TERMINAL WINDOW

The ***Terminal Window*** section of the LIN Message Center displays all transactions between the LIN Message Center PC program and one or more LIN/RS-232 Converters. It also shows other traffic on the LIN bus, including any other LIN-compatible devices currently active on the LIN bus. Each transaction is time-stamped.

10.5. NODE LIST

The ***Node List*** window in the LIN Message Center screen displays the connected LIN/RS-232 Converter(s). Each is identified by the PC serial port to which it connects. In the above example screen, there are two LIN/RS-232 Converters, connected to serial ports COM1 and COM2 on the PC.

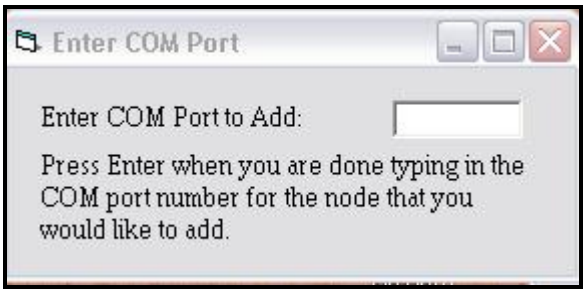
The Node List does not show other active LIN-compatible devices (that are not LIN/RS-232 Converters), because the PC software does not need to control them. However messages to and from these devices appear in the Terminal Window.

For certain applications—for example, end-of-line testing of multiple identical ECUs—it is convenient to connect multiple LIN/RS-232 Converters to the same PC. Most PCs support only one or two RS-232 serial ports, but for PCs supporting USB (Universal Serial Bus), additional serial ports can readily be added using USB to serial converters. *(Contact Silicon Engines for assistance on such a system.)* There can be up to 16 LIN/RS-232 Converters attached at any one time.

10.6. ADDING A NEW NODE

When first installing the LIN Message Center and one or more LIN/RS-232 Converters, it is necessary to configure the PC software so as to recognize each LIN/RS-232 Converter. This is called *adding a new node.*

To add a new node, click on the ***Add a Node*** button. You will see a window as follows:



ADDING A LIN/RS-232 CONVERTER NODE
FIGURE 10.6.1.

In this window, click on the entry box to the right of the text that says *Enter COM Port to Add* and enter a COM port number. For example, for the standard serial port COM1, type ***1*** and then press ***ENTER***. After any number that you type, be sure to press ***ENTER***. The LIN Message Center program will try to add that particular COM port to the node list. If it could not add that port, the box will turn RED. The box will also turn red if a previously initiated COM port later becomes unavailable. This could happen if a COM port has been disabled or removed under the Windows Control Panel. Sometimes a USB-to-serial COM port will not show up if its driver software has not yet been installed.

10.7. REMOVING A NODE

To remove a node that has already been added, go to the ***Node List*** window and click on the node you want to remove. That node should then become highlighted—a gray box will appear over that node. Click on the ***Remove Selected Node*** box. This will remove the selected node.

10.8. THE SELECTED NODE

At all times one of the nodes in the ***Node List*** window will be selected. To change the selection, go to the ***Node List*** window and click on the node that you want to select. A majority of LIN Message Center commands will be sent specifically to the selected node—the node that was most recently highlighted in the ***Node List*** window.

10.9. RESIZING THE TERMINAL WINDOW

If you want to increase the size of the ***Terminal Window***, drag the lower right corner of the LIN Message Editor window down and to the right. The Terminal Window lower right corner will follow the lower right corner of the LIN Message Editor window.

10.10. QUERY STATUS

The ***Query Status*** button in the LIN Message Center will automatically send QUERY STATUS commands to all connected LIN/RS-232 Converters. Each node will report the most recent error condition (if applicable) as well as the software revision of that node.

10.11. CLEAR BUFFERS

The ***Clear Buffers*** button is only necessary if a buffer overflow error condition occurs. This will not happen very frequently but could happen under high bus utilization situations. The CLEAR BUFFERS command is sent to whichever node has most recently been selected on the Node List.

10.12. SEND WAKE-UP

The ***Send Wake-Up*** button in the LIN Message Center is used to send a bus wake-up message.

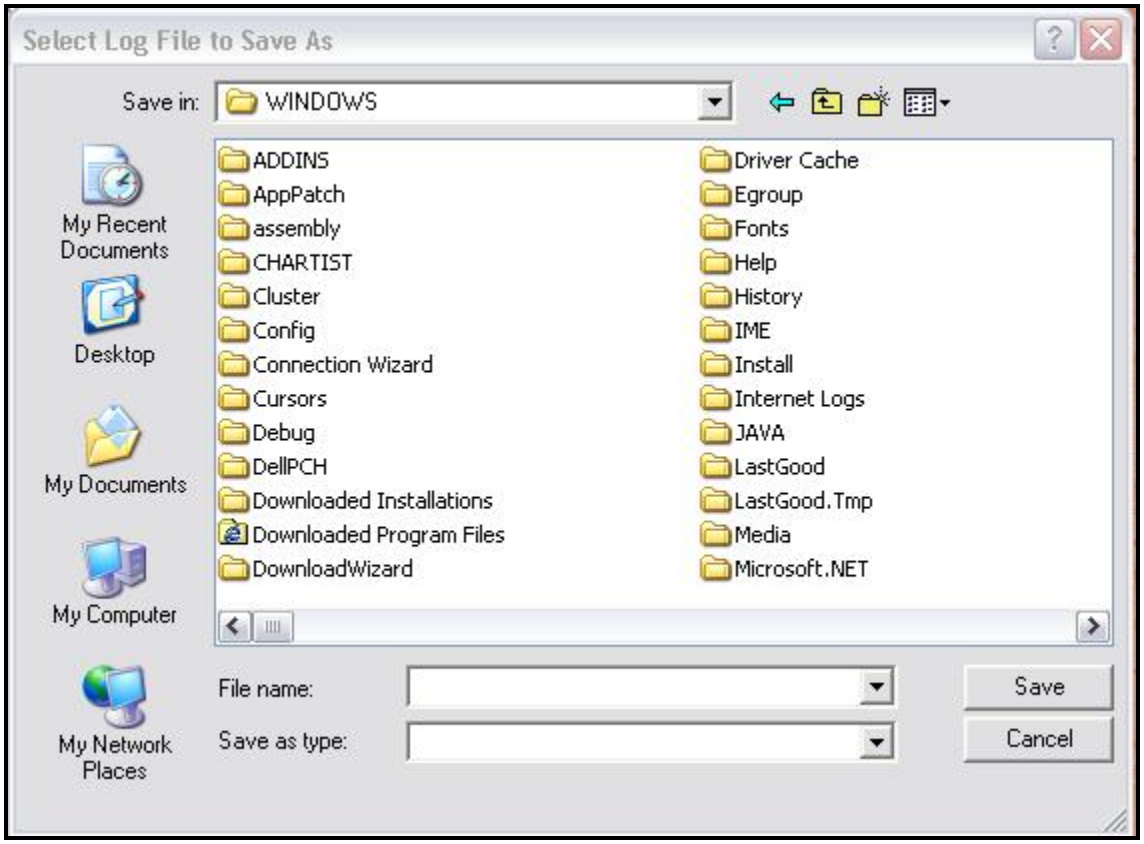
If bus wake-ups have been disabled through the ***Configure LIN Box*** menu *(Sec. 9.15)*, then the ***Send Wake-Up*** button in the LIN Message Center will be grayed out, and will be inactive.

If bus wake-ups are enabled, then the LIN/RS-232 Converter will send a bus wake-up message over the LIN bus. If no LIN bus traffic is detected in response, the LIN/RS-232 Converter tries again. After three tries of sending a wake-up signal, the LIN/RS-232 Converter waits a specified time (LIN 1.3: 15,000 bit times; LIN 2.0: 2.5 seconds), and rechecks to see whether continuous wake-ups are enabled. If continuous wake-ups are enabled then the Converter automatically retries the wake-up sequence.

When bus wake-ups are enabled, when you press the ***Send Wake-Up*** button on the LIN Message Center screen, you should see the DSR lamp on the LIN/RS-232 Converter turn GREEN. If continuous wake-ups are enabled, the DSR lamp will stay green. If the continuous wake-up option is NOT enabled, the DSR lamp will change to red after a short period.

10.13. START LOGGING

The ***Start Logging*** button in the LIN Message Center is used to enable automatic recording of the messages in the Terminal Window to a file on the PC. You will see a window as follows:



SELECTING A LOG FILE NAME
FIGURE 10.13.1.

Use the Windows Explorer functionality to select the location for the log file.

Under ***File name:*** specify the name and extension of the log file.

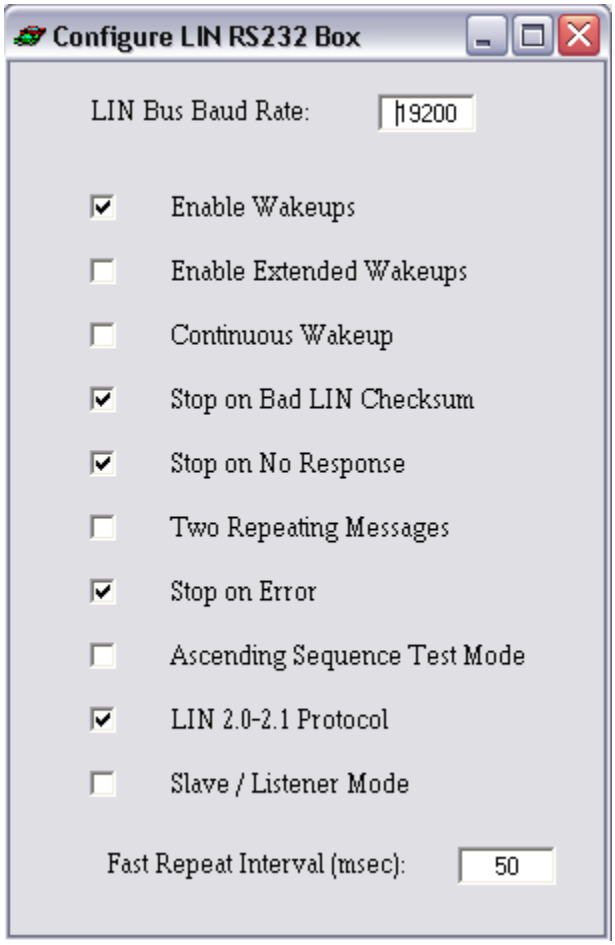
Under ***Save as type:*** you can specify any desired file type. But no matter how specified, the log file will be saved as an ASCII text file, with a new line sequence at the end of each message.

10.14. STOP LOGGING

The ***Stop Logging*** button in the LIN Message Center terminates automatic recording of the messages in the Terminal Window. The log file is closed.

10.15. CONFIGURE LIN BOX

The configuration parameters for the LIN/RS-232 Converter are not complicated. Click on ***Configure LIN Box*** and the following screen will pop up:



CONFIGURE LIN BOX SCREEN
FIGURE 10.15.1.

The ***LIN Bus Baud Rate*** can be anywhere from 1,000 to 20,000. This is the speed of the LIN traffic.

The second option is the ***Enable Wakeups*** option. Click the check box if you want to allow the LIN/RS-232 Converter to generate LIN bus wake-up commands.

The ***Continuous Wakeup*** option programs the LIN/RS-232 to generate continuous bus wake-up commands. This button will be grayed out (inactive) if the ***Enable Wakeup*** button above it is not checked.

The ***Stop on Bad LIN Checksum*** option causes the LIN Message Center to stop sending any repeated messages if it ever detects a bad LIN checksum at the end of a frame. This functionality can be useful for ECU testing.

The ***Stop on No Response*** option causes the LIN Message Center to stop sending any repeated messages if it ever detects that an ECU did not respond to a LIN message. This functionality can be useful for ECU testing.

The ***Two Repeating Messages*** option modifies the ***Send Selected Message Repeatedly*** function on the LIN Message Center screen. Instead of sending just a single selected message repeatedly *(Sec. 9.19)*, the LIN/RS-232 Converter sends both the selected message, and the message immediately preceding it in the Message Edit Window *(Sec. 9.16).*

The ***Stop on Error*** option causes the LIN Message Center to stop sending any repeated messages if it detects any error. *(See the error list in Figure 8.6.2*.) This functionality can be useful for ECU testing.

The ***Ascending Sequence Test Mode*** option modifies the ***Send Selected Message Repeatedly*** function on the LIN Message Center screen. Instead of sending just a single selected message repeatedly *(Sec. 9.19)*, the LIN/RS-232 Converter sends a continuous stream of LIN messages, from 00 hex to 3F hex, then repeating. The LIN/RS-232 Converter automatically appends the two most significant LIN parity bits to the ID byte, and the packet checksum. The sequence continues until the ***Send Selected Message Repeatedly*** function is turned off, or until an error is encountered (providing that one of the boxes is checked that causes the LIN Message Center screen to stop on a detected error).

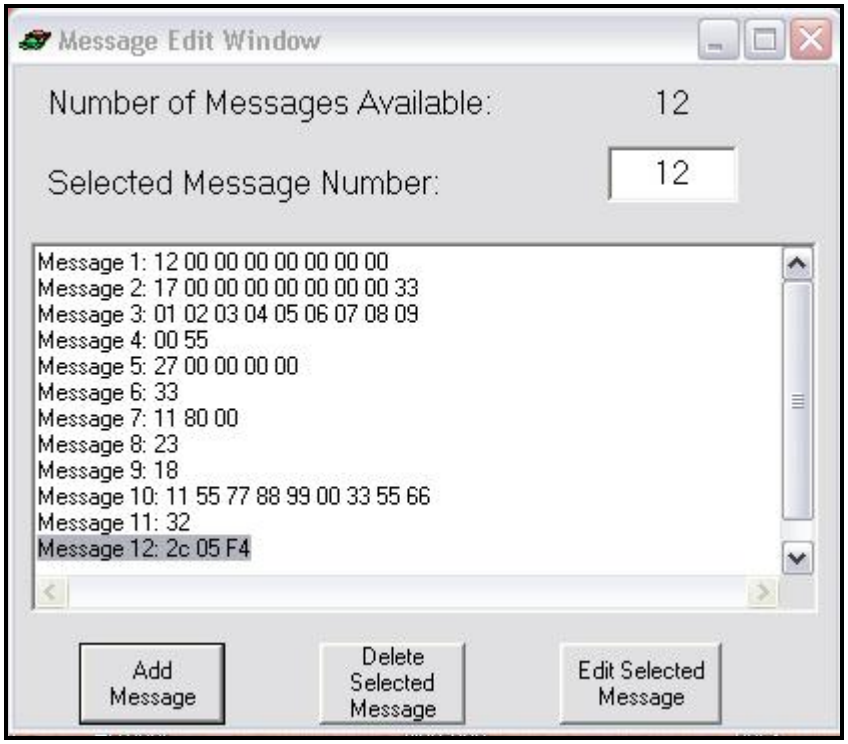
The ***LIN 2.0 Protocol*** button, when checked, causes the LIN/RS-232 Converter to operate according to the LIN 2.0 protocol. When not checked, the LIN/RS-232 Converter operates in LIN 1.3 mode.

The *Slave/Listener Mode* option, when checked, causes the LIN/RS232 Converter to place a Slave Mode bus resistance on the bus instead of the default Master Mode bus resistance.

The *Fast Repeat Interval (msec)* can be set anywhere from 0 to 5000. This is in units of milliseconds. This specifies the time between messages being sent in Fast Repeat mode.

10.16. EDITING LIN MESSAGES

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



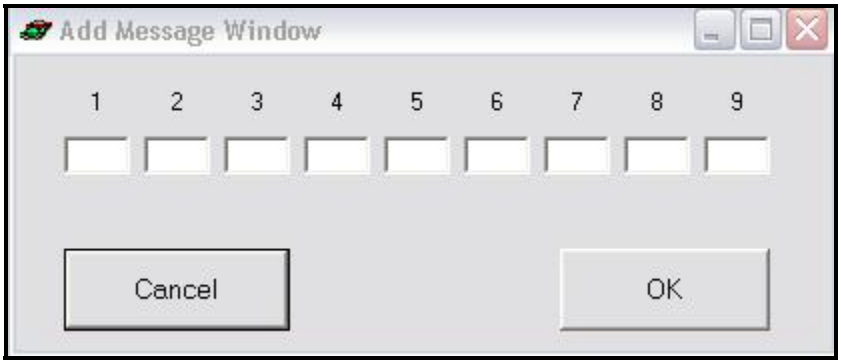
MESSAGE EDIT WINDOW
FIGURE 10.16.1.

To select a message, click on the message you would like to send next or to remove, just as in the ***Node List*** window. 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***.

10.17. ADDING A NEW LIN MESSAGE

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



ADD MESSAGE WINDOW
FIGURE 10.17.1.

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

If you enter just one hex number in box 1, then this will be a master mode message with a response that will be supplied by a slave device on the LIN bus.

If you fill more than one hex number in a box, then this will be a complete LIN message. Note that the LIN checksum is computed automatically and should not be included in the data entered here.

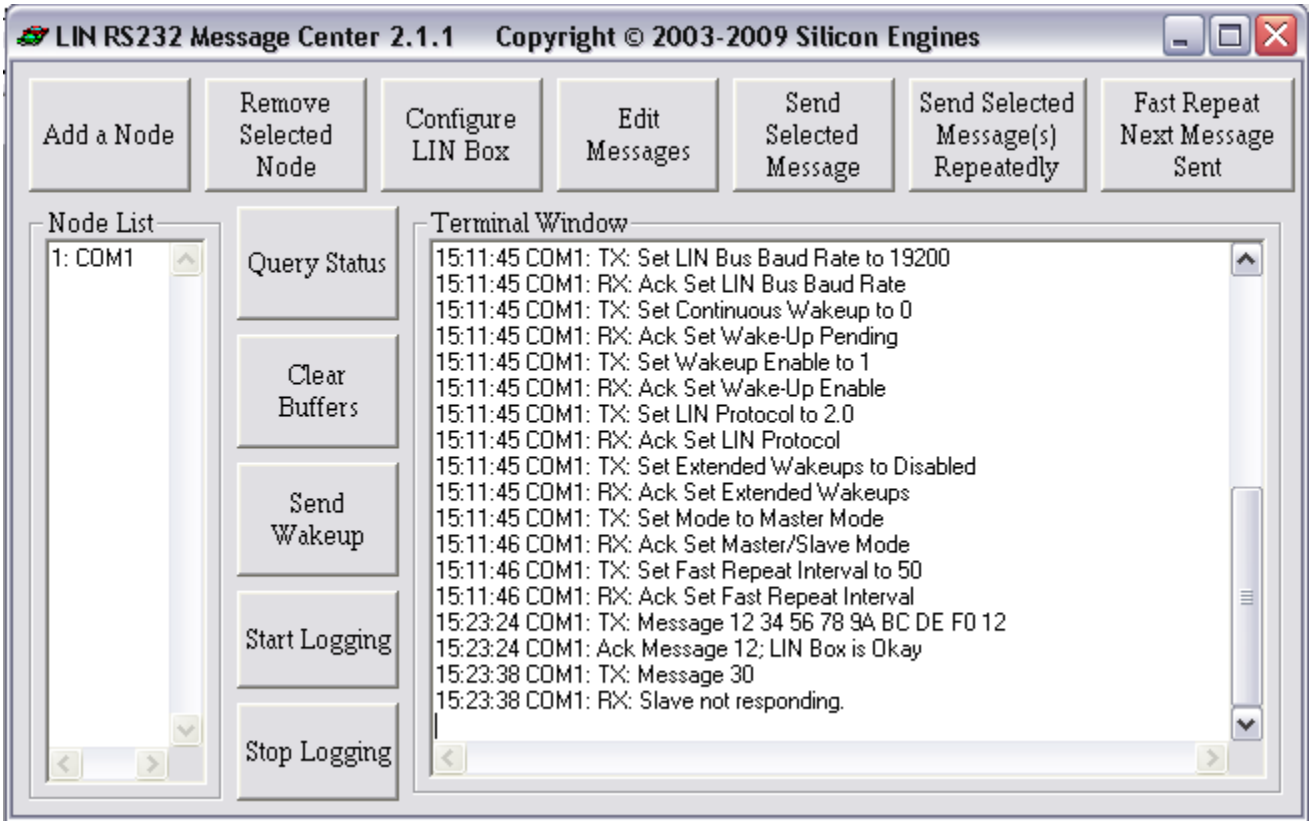
Note also that the first byte must have bits 6 and 7 cleared. In other words, the first byte must be less than or equal to 3F hex. The two high-order bits (parity bits) will be filled in automatically.

When you are done entering the message, press the ***OK*** button and the message will appear in the Message Edit window *(Fig. 10.16.1)*.

Note that in Fig. 10.16.1, messages 6, 8, 9, and 11 are master mode messages, while the others are complete master/slave messages.

10.18. SENDING A LIN MESSAGE

To send the most recently selected message in the ***Message Edit Window***, press ***Send Selected Message*** on the main screen. The following screen shows an example:



EXAMPLE OF SENDING A MESSAGE
FIGURE 10.18.1.

In this ***Terminal Window*** example, message 12 was sent at 15:23:24, containing eight data bytes as part of the master message. The next line is a message generated by the LIN/RS-232 Converter indicating that the message was transmitted correctly. Message 30 was sent at 15:23:38, and there was no slave response. The LIN Message Center software automatically checks the LIN checksum and reports whether it was OK or not. It does not display the LIN checksum itself, although this information is available to PC programs.

10.19. SENDING A LIN MESSAGE REPEATEDLY

You can send a message repeatedly by clicking the ***Send Selected Message Repeatedly*** button on the main LIN Message Center screen. This will cause the selected message to be sent to the selected nodes over and over.

If you have checked the ***Two Repeating Messages*** option in the Configure LIN Box menu *(Sec. 10.15)*, then both the selected message, and the message immediately preceding the selected message in the Message Edit Window, will be sent repeatedly.

There are options in the Configuration Screen that halt the repeating messages when certain errors are encountered—bad checksum, no response, etc. These options are useful for testing the LIN capabilities of an ECU on the LIN bus.

10.20. SENDING REPEATED MESSAGE WITH FAST REPEAT FUNCTIONALITY

You can send a single message at a very fast repeat rate using the fast repeat functionality. To select the fast repeat rate, select an interval in milliseconds between 0 and 5000 from the Configuration screen. When you want to fast repeat your message, click the button labeled Fast Repeat Next Message Sent. Then, the next message that you send with the Send Selected Message button will be sent repeatedly at the fast repeat interval until a Clear Buffers command is sent. Note that if the message is a full message (master plus slave task; master ID and payload), then there will be no acknowledgements reported to the PC, but if the message is just a master task where the slave task from a slave device fills in the rest of the message, then the slave's response bytes WILL be reported back to the PC. Also keep in mind that while in Fast Repeat mode, you cannot send any more configuration or Send Message commands until a Clear Buffers command is sent.

11. TROUBLESHOOTING

11.1. EXCESSIVE BUS CAPACITANCE

Communications failures can result when there is excessive bus capacitance on the LIN network. *(See Sec. 2.11 above for specifications.)*

Electrically the LIN bus is a wired-OR network. An open-collector bipolar transistor (or open-drain MOSFET) actively pulls the bus down to generate a low level on the bus. However passive elements are relied upon to return the bus to a high level—the 1 KΩ resistor at the master node, and the 30 KΩ resistors at the slave nodes.

Excessive capacitive loading delays the low-to-high level transitions on the bus, potentially causing data errors.

Users of the LIN/RS232 Converter encountering LIN data errors should check the LIN bus line with an oscilloscope for excessive low-to-high bus transition times, approaching or exceeding one-half bit time at the effective bit rate.

11.2. BUS RESISTIVE LOADING

In the event that excessive capacitive loading is suspected, as an experimental counter-measure, try adding a 10 KΩ external pull-up resistor between the LIN bus line and VBATT. This resistor is effectively in parallel with the 1 KΩ pull-up resistor within the LIN/RS-232 Converter, and the 30 KΩ pull-up resistors at the slave nodes, and tends to reduce the low-to-high bus transition time. Recheck the bit transition times on the oscilloscope.

You can also try changing the Master/Slave Mode using a command to the LIN/RS-232 Converter to switch in either the Master Mode load resistor (1 KΩ) or the Slave Mode resistor (30 KΩ).

From the standpoint of the LIN/RS-232 Converter, the total effective bus resistance—including the 1 KΩ resistor within the LIN/RS-232 Converter, the 30 KΩ resistors at all connected slave nodes, and the external pull-up resistor—can be as low as 250 Ω, with VBATT at 14 VDC or lower. The current-limiting circuit within the LIN/RS-232 Converter protects its transmitter circuitry from excessive bus loading. The current limit is approximately 58 mA, but is not tightly controlled. If the total effective bus resistance falls too low—or VBATT is significantly higher—then the transmitter circuit will go into current-limit when it is driving the LIN bus low. No harm will be done to the LIN/RS-232 Converter, but the LIN bus voltage will no longer be pulled fully down to 0 V. As the total effective bus resistance falls still lower, eventually the LIN/RS-232 Converter will no longer be able to generate valid low-level bus signals.

From the standpoint of a connected ECU, the minimum effective bus resistance, before its transceiver reaches current limit, depends on the LIN bus transceiver circuitry used in that product.

12. CUSTOM SOFTWARE

12.1. CUSTOM PC SOFTWARE

Please contact Silicon Engines if you are interested in customized PC software for a particular LIN system.

12.2. CUSTOM ECU SOFTWARE

Please contact Silicon Engines if you are interested in customized embedded microcontroller software for an automotive or industrial ECU running the LIN protocol. See also Part 7 for details on variant firmware versions that already are available.

13. REFERENCES

13.1. LATEST VERSIONS

The documents shown below were the latest revisions known when this document was last revised. They are generally available from the issuing organization's website. Please be sure to check with the issuing organization for updates and revisions.

13.2. LIN SPECIFICATIONS

1. *LIN Specification Package,* Revision 2.1, pages 1-27, November 24, 2006. Introduction to the LIN specification package. LIN Consortium.
2. *LIN Protocol Specification,* Revision 2.1, November 24, 2006, pages 23-53. Specifications for LIN data link layer. LIN Consortium.
3. *LIN Transport Layer Specifications,* Revision 2.1, November 24, 2006, pages 24-62. Specifications for ISO 15675-2-compatible diagnostic messages. LIN Consortium.
4. *LIN Node Configuration and Identification Specification,* Revision 2.1, November 24, 2006, pages 63-76. Specifies the manner in which a slave node is configured and identified. LIN Consortium.
5. *LIN Diagnostic Specification,* Revision 2.1, November 24, 2006, pages 77-105. Specifies communications between a diagnostic tester and slave nodes. LIN Consortium.
6. *LIN Physical Layer Specification,* Revision 2.1, November 24, 2006, pages 106-123. Specifications for LIN bit timing and voltage levels. LIN Consortium.
7. *LIN Application Program Interface Specification,* Revision 2.1, November 24, 2006, pages 124-160. Specifies a recommended application programmer's interface between the LIN software and ECU application software. LIN Consortium.
8. *LIN Node Capability Language Specification,* Revision 2.1, November 24, 2006, pages 161-171. Specifies a means to characterize an off-the-shelf LIN slave node in machine-readable syntax as a step towards plug-and-play operation.
9. *LIN Configuration Language Specification,* Revision 2.1, November 24, 2006, pages 172-191. Specifies a means to characterize a LIN network, facilitating simulation in the absence of one or more nodes.

13.3. LIN APPLICATION NOTES

1. *Software LIN Slave,* Application Note AVR308, May 2002, 12 pages. Atmel Corporation.
2. *LIN Bus and its Potential for Use in Distributed Multiplex Applications,* SAE Technical Paper 2001-02-0072, 2001, 10 pages. Delphi Automotive Systems; Society of Automotive Engineers.
3. *LIN Protocol Specification Implementation with PICmicro® MCUs,* Application Note 729, DS00729A, 2000, 35 pages. Microchip Technology Inc.
4. *Local Interconnect (LIN) Demonstration,* Application Note 2103, 2000, 68 pages. Freescale Semiconductor (formerly Motorola Semiconductor Products).
5. *Philips Microcontrollers in LIN Applications,* Application Note 10115, Feb. 15, 2002, 6 pages. Philips Semiconductors.
6. *LIN (Local Interconnect Network) Solutions,* Application Note 1278, Rev. 1.1, April 2002, 44 pages. STMicroelectronics.

13.4. LIN TRANSCEIVERS

1. Freescale Semiconductor, MC33399.
2. Infineon Technologies, TLE6258.

3. Melexis Microelectronic Integrated Systems, TH8080.
4. ON Semiconductor, NCV7310.
5. Philips Semiconductors, TJA1020.
6. STMicroelectronics, L9638.

13.5. WEBSITES

1. Atmel Corporation, www.atmel.com
2. Delphi Automotive Systems, www.delphi.com
3. Freescale Semiconductor, www.freescale.com
4. Infineon Technologies, www.infineon.com
5. LIN Consortium, www.lin-subbus.de
6. Melexis Microelectronic Integrated Systems, www.melexis.com
7. Microchip Technology Inc., www.microchip.com
8. ON Semiconductor, www.onsemi.com
9. Philips Semiconductors, www.semiconductors.philips.com
10. Society of Automotive Engineers, www.sae.org
11. Silicon Engines, www.siliconengines.net
12. STMicroelectronics, www.st.com

14. REVISION HISTORY

14.1. REVISION H

1. **Throughout:** Corrected cross-references between sections. Updated references to LIN specifications to version 2.1.

2. **Sec. 1.3:** Added information on compatibility with LIN 2.1 and information on pass-through mode. Added note that 19,200 baud pass-through mode is supported.Added extra feature information to wakeup section for extended wakeups. Added extra information regarding synchronization break length.

3. **Added Sec 7:** Firmware variants description.

4. **Sec. 8.14:** Added information on compatibility with LIN 2.1.

5. **Sec. 8.15:** Added extended wakeup details.

6. **Sec 8.16:** Added master/slave mode details.

7. **Sec 8.17-8.18:** Added fast repeat mode details.

8. **Sec. 8.19:** Added pass-through mode details.

9. **Fig 8.4.1:** Added pass-through mode command, master/slave mode command, fast repeat commands, and extended wakeup command.

10. **Fig. 9.6.2:** Added details about error code 22H related to fast repeat mode.

11. **Fig 10.3.1:** New picture of main screen.

12. **Sec 10.20:** Added section on Fast Repeat Mode from Message Center.

13. **Sec 10.15:** Added details for Slave/Listener mode check box and Fast Repeat Interval text box.

14. **Fig 10.15.1:** New picture of configuration screen.

15. **Fig 10.18.1:** New picture of send message example.

16. **Sec 10.18:** New explanation of send message example.

17. **Sec 11.2:** Added note that you can also use the Master/Slave mode setting command to aid in setting the bus pull-up resistance.

18. **Sec 12.2:** Added note to see section 7 for firmware variants that already exist.

14.2. REVISION G

1. **Sec. 9.1:** Added information on compatibility with Windows versions.

14.3. REVISION F

1. Updated Silicon Engines address and contact information.

14.4. REVISION E

1. **Throughout text:** Updated references from LIN Protocol Specifications Rev. 1.3, to applicable section in Rev. 2.0.
2. **Sec. 2.11**: Added this section on capacitive bus loading.
3. **Sec. 6.4**: Corrected lamp colors for signal CTS (red and green were reversed).
4. **Sec. 7.4:** Added command 94H to chart.
5. **Sec. 7.11:** Changed recommended non-zero argument from FFH to 01H for consistency with other commands. (FFH will still work properly.) Under *Note 2*, clarified that when set for LIN 2.0, timing is set to specified time periods, not based on bit times.
6. **Sec. 7.13:** Corrected section heading to show command code is 93H. Changed recommended non-zero argument from FFH to 01H for consistency with other commands. (FFH will still work properly.)
7. **Secs. 9.1, 9.2:** Modified to show Message Center software is now provided on a CD rather than on floppy diskettes.
8. **Sec. 9.3:** Provided an updated message center screen image.
9. **Sec. 9.5:** Updated to show there are two active nodes in the example message center screen.
10. **Sec. 9.12:** Updated name of button to SEND WAKE-UP. Updated to show delay timing differs depending on whether unit is operating in LIN 1.3 or LIN 2.0 mode.
11. **Sec. 9.15:** Updated configuration screen, added paragraph on LIN 2.0 button.
12. **Sec. 9.16:** Updated message edit window, adjusted descriptive text to match window.
13. **Sec. 9.17:** Updated the add message screen, corrected to provide nine windows (instead of 10).
14. **Part 10:** Added this part on troubleshooting.
15. **Secs. 12.4, 12.5:** Changed from Motorola to Freescale Semiconductor.

14.5. REVISION D

1. **Throughout text:** Updated all references to the LIN Protocol Specifications Rev. 1.2, to correct section in Rev. 1.3.

2. **Sec. 5.2:** Added typical supply current specification.
3. **Sec. 7.4:** Renamed the 91H command as CONTINUOUS WAKEUPS. Added 93H command to the list.
4. **Sec. 7.8:** Added note that the 82H command will be ignored if wake-up signals have been disabled by the 93H command.
5. **Sec. 7.11:** Renamed the 91H command as CONTINUOUS WAKEUPS. Clarified that this function can be disabled through the 93H command.
6. **Sec. 7.13**: Added the 93H command.
7. **Sec. 9.4:** Added separate section on the Terminal Window.
8. **Sec. 9.5:** Added separate section on the Node List. Clarified that the Node List includes only the LIN/RS-232 Converters, not other LIN-based ECUs.
9. **Sec. 9.6:** Clarified the purpose of the ADD A NEW NODE function.
10. **Sec. 9.12:** Added wording to clarify that the WAKE-UP BUS button may be disabled.
11. **Secs. 9.13, 9.14:** Added these sections on log files.
12. **Sec. 9.15:** Rewrote to include descriptions of ENABLE WAKEUPS, STOP ON NO RESPONSE, TWO REPEATING MESSAGES, STOP ON ERROR, and ASCENDING SEQUENCE TEST MODE.
13. **Sec. 9.19**: Added wording regarding the TWO REPEATING MESSAGES option.

14.6. REVISION C

1. **Cover and Sec. 1.1:** Added Model 9003 designation to this document.

14.7. REVISION B

1. **Effective software release:** This change is effective as of firmware revision 4.
2. **Sec. 8.4, LIN Message Packet:** Changed to show that Length Indicator can have a value of 88H.

14.8. REVISION A

Initial release.

☑