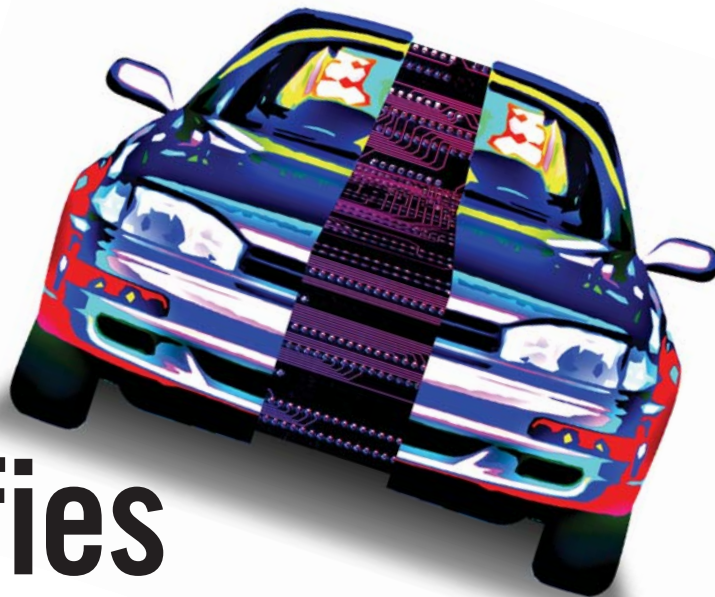


THE GROWTH OF ELECTRONIC SUB-SYSTEMS IN VEHICLES DEMANDS NETWORKS FROM THE MISSION-CRITICAL TO THE MUNDANE. THE LOCAL-INTERCONNECT-NETWORK PROTOCOL TACKLES THE TASK.



LIN simplifies and standardizes IN-VEHICLE NETWORKS

WITH PREDICTIONS suggesting a global annual-compound-growth rate of some 8.3% over the next five years, the automotive sector continues to be our industry's fastest growing market. It's sobering to consider that, barely 20 years ago, the sole electronics content within most vehicles was a radio. Then came the electronic ignitions, engine-management units, and antilock-braking systems that are standard in today's entry-level cars. Now, of course, vehicles include previously unheard-of luxuries, and top-of-the range vehicles adopt sophisticated electronics, such as intelligent cruise controls that automatically maintain a safe distance from the vehicle ahead. Estimates suggest that the electronics content of an average car now accounts for no less than 22% of its manufacturing cost, creating markets for the embedded controllers, power devices, and communications technologies that link its ECUs (electronic-control units).

Recognizing the need for a robust in-vehicle network to manage distributed intelligence and reduce wiring-harness dimensions, Bosch in 1986 designed a CAN (controller-area network). Today, CANs dominate in-vehicle networking and have also made the transition to multiple industrial uses. In the meantime, other networking systems have appeared to tackle emerging automotive applications. These technologies include D2B (domestic-digital-databus), FlexRay, and MOST (media-oriented system transport), all of which employ fiber media for speed and EMC resistance. TT-CAN (time-triggered extensions to CAN) improve the protocol's determinism, as well as the TTP (time-triggered-protocol) series that competes with FlexRay for safety-critical use (**Reference 1**). Because these systems serve high-

end applications and are relatively expensive, designers require a low-cost alternative to serve mundane tasks, such as to control body functions from seats to sunroofs. As a result, car makers increasingly embrace LINs (local-interconnect networks), which position themselves at the lowest level in the automotive-networking hierarchy (**Figure 1**).

The first LIN specification appeared in 1999. Among the founding members of the LIN Consortium, its design authority, are car makers BMW, DaimlerChrysler, Volkswagen Audi Group, and Volvo Cars, together with hardware and networking expertise from Freescale Semiconductor and Volcano Automotive Group. Design influences include the Vlite bus that several car makers use, as well as lessons accruing from many years of CAN evolution and development. Several amendments culminated in LIN Version 1.3 in November 2002, which many observers regard as the first stable release. Further work resulted in a major revision that appears as the current Version 2.0 of September 2003, which the LIN Consortium recommends for all new development.

Meanwhile, in North America, the Society of Automotive Engineers issued its J2602 recommended practice, "LIN Network for Vehicle Applications," with key car-maker representation coming from Ford and General Motors. The main differences between LIN 2.0 and J2602 include limiting the transmission rate to 10.4 kbps and modifying some protocol details, such as error handling. Some observers feel that J2602's objectives include limiting feature creep, thus making it easier to meet LIN's overriding low-cost target using, for example, state-machine logic rather than microcontroller-based intelligence.



The LIN Consortium's objectives for its new system don't stop with low cost, although, because LIN is markedly cheaper than CAN or the J1850 domestic-US standard, cost is the prime driver. In fact, with original predictions for LIN lying at around \$1/node, it is currently proving hard to meet this cost target. But, in its primary role as a sub-bus, LIN's design ensures that it functions as a logical and scalable extension of CAN and J1850. It provides acceptable reliability for non-safety-critical tasks, with less-than-100-msec response times and predictable worst-case timing characteristics. Learning from previous bus evolutions, the developers were also careful to consider tool-chain-support issues. Such considerations have become crucially important as the car makers forge seamless co-development links with their system suppliers.

Naturally, the specification must ensure hardware and software interoperability among multiple vendors, as well as minimizing peripheral but critical issues, such as EMC. From the outset, LIN's specifications accordingly subdivide into three main parts that describe the transmission medium and its communication protocols, a configuration language, and APIs (application-programming interfaces) (Figure 2). Representing the lowest two levels of the ISO/IEC 7498-1:1994 open-systems-interconnect model, the protocol specification tackles the physical-layer and data-link-layer mechanisms. At the highest level, an API abstracts the user's code from lower level network mechanics; in between, a signal interaction and diagnostic layer decouples the application from the network. To furnish a standard interface between LIN nodes from mul-

AT A GLANCE

- ▷ LIN (local-interconnect network) challenges CAN (controller-area network) for lowest cost in noncritical applications.
- ▷ Version 2.0 introduces new plug-and-play capability but costs memory.
- ▷ Vendors offer a dazzling palette of hardware support.
- ▷ Future growth prospects include consumer goods and industrial use.

iple suppliers, the LIN configuration-language description defines the format of the files that configure the network. These configuration files also provide hooks into development tools. In a further and major forward step, 2.0 introduces the LIN-node-capability language, easing integration via a plug-and-play concept (see sidebar "LIN 2.0 goes plug and play" on the Web version of this article at www.edn.com).

ONE-WIRE MASTER/SLAVE ARCHITECTURE

To minimize cost and wiring weight, LIN uses a single-conductor, wire-OR bus that takes advantage of a car's body shell to serve as a common ground. Each LIN subnet comprises one master and at least one slave node to a maximum of 16 devices per bus. Nodes can participate on more than one LIN bus, and masters may also operate as bridges into other network environments, typically CANs. Maximum transmission speed and reach are 20 kbps and 40m, respectively, using UART/SCI communications. This technology makes LIN implementations pos-

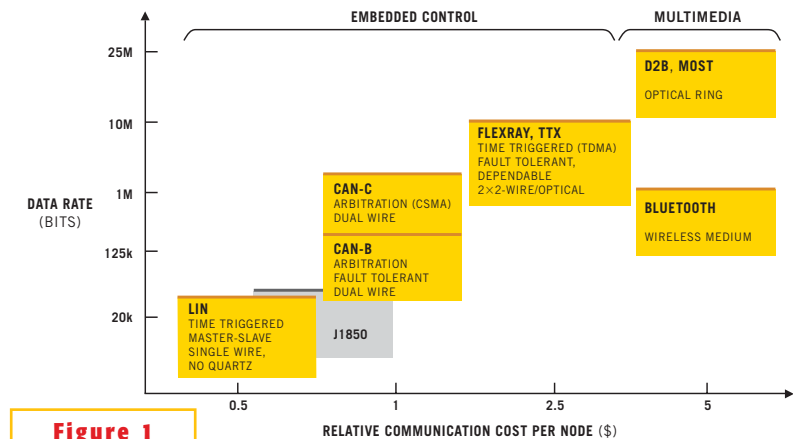


Figure 1

LIN tackles the lowest level of the in-vehicle-networking hierarchy.

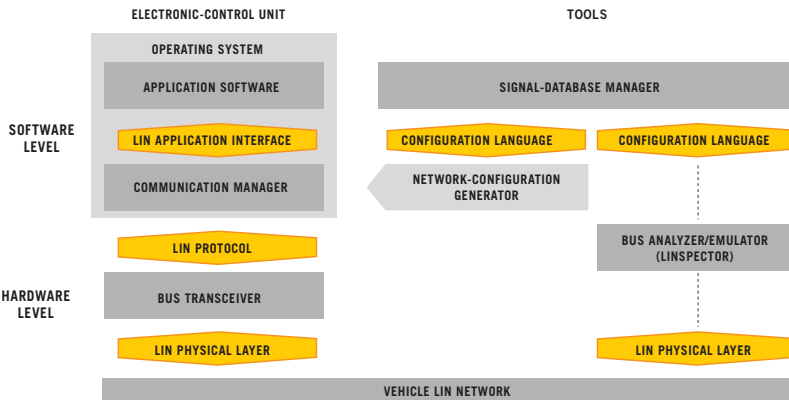


Figure 2 LIN defines the physical-layer, protocol-handler, and application-programming interfaces with support from a configuration language.

sible with drivers ranging from simple state-machine logic to “bit-bashing” an I/O pin in software to using serial peripherals. Such low speeds also constrain interference generation and ease timing issues, assisted by a master-driven self-synchronization facility that allows slave nodes to dispense with crystal or resonator timers.

Physical-layer exchanges employ an enhancement to the ISO-9141 standard that dominates European and Japanese vehicle-diagnostic systems. LIN-compatible line drivers limit slew rate to around 2V/ μ sec to avoid creating interference due to fast edges. Such a line driver asserts the “dominant” logic-zero state by driving the bus line to within 20% of system ground; a “recessive” logic one requires driving the line to within 20% of battery voltage. To allow for effects such as ground shift, receivers allow more tolerance, acknowledging levels within 40% of the respective rails (Figure 3). The master terminates the bus with a 1-k Ω resistor to battery voltage, and each slave defaults its I/O line high with a 30-k Ω pullup resistor. A diode in series with the termination resistor prevents devices on the bus from backfeeding into the battery-voltage rail if the supply fails. Masters and slaves also each present around 220 pF to the line up to a maximum of 10 nF per bus, which results in a system time constant of 1 to 5 μ sec. In a region in which AM radios still proliferate, SAE J2602’s 10.4-kbps limitation further eases compatibility issues for the North American market.

Unlike CAN, LIN’s master/slave architecture avoids data-traffic collisions and the need for arbitration logic by having

the master supervise message transmissions, thus ensuring that only one message transmits at any time. A frame consisting of the master’s header, a pause, and a slave’s response encapsulates each message exchange (Figure 4a). Start and stop bits surround each byte, resulting in a 10-bit transmission per byte. There are several frame types, starting with diagnostic frames that carry diagnostic or initialization information. By contrast, “unconditional frames” always carry signals of as many as eight bytes. These are the frame types typify applications. “Sporadic” frames also always carry signals, but slaves respond only if new data is available, otherwise leaving the data field blank—an attempt to add some dynamic behavior into the system’s schedule without compromising its determinism. Polling infrequently responding nodes generates bus traffic. To improve system responsiveness by reducing the bus traffic, the protocol includes an event-triggered frame. This frame accommodates as many as seven data bytes, because the first field carries an identifier that associates the frame with its task. Again, slaves respond only if they have new data. The protocol also provides

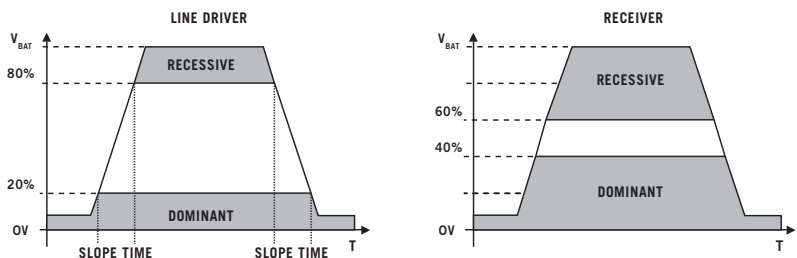


Figure 3 Receivers tolerate levels of as much as 40% of the supply rails to account for effects such as ground shifts.

for user-defined frames and reserves another type for future use.

To initiate a data transfer following an interframe space or bus-idle condition, the master transmits a header comprising a synchronization-break period, a single-byte-synchronization field, and an identifier byte. The identifier byte carries six bits of information and two parity bits, allowing 64 message identifiers (Figure 4b). In normal operation, there is no addressing as such; rather like CAN, the identifier byte uniquely defines the purpose of the frame. Identifier decimal values of zero to 59 carry signals; 60 and 61 are master-request and slave-response diagnostic frames, respectively; user-defined frames have a header value of 62; and 63 is reserved. Each slave waits for the synchronization break and locks onto the synchronization byte before scanning the bus message. One or more slaves then receive data, or a single slave transmits response data. The data field accommodates eight bytes; the data field’s association with its identifier byte predefines the field’s length. A single-byte check field terminates the transmission, providing an error-detection facility. The master is responsible for all error handling, which is in turn the application programmer’s responsibility; LIN 2.0 has no defined error-handling mechanism.

Because it’s imperative to preserve battery power when the vehicle is inoperative, slaves automatically enter sleep mode if the bus is idle for more than four seconds. The master can also force slaves into sleep mode by sending the diagnostic master-request frame with the first data byte set to zero. The master subsequently monitors the bus when it is idle, looking for wake-up signals from slaves that require service. Any bus node can request wake-up by asserting the dominant state for 250 μ sec to 5 msec, which makes 5 msec the dominant state’s longest valid



run. Most transceivers incorporate a watchdog timer that disconnects nodes exceeding this maximum, because such behavior indicates an error condition that would otherwise monopolize the bus. Slaves must wake up and be ready to transmit data within 100 msec of the end of the wake-up signal. Crucially, because the message length, interframe-spacing parameters, and device-wake-up times are known, it's easy to calculate worst-case response times for any system. Masters typically use a static, round-robin scheduler, although adaptive schedulers provide greater flexibility to permit decision-based systems that similarly have guaranteed determinism.

An exception to the zero-collision rule occurs when the master polls for an event-triggered frame and more than one slave responds within the same time slot. This situation might arise, for example, when the master polls all the doors within a central-locking application using an event-triggered frame. The response would normally be blank, but if more than one door button is active at this instant, more than one slave responds. The master resolves the collision by requesting all of the unconditional frames of similar association and checks their event flags before again requesting the event-triggered frame. This sequence avoids the possibility of a slave's withdrawing from a collision without corrupting the data, which the master would not detect, and thus lose the slave's response. Because application software implements these sequences, the programmer must ensure that the bus has enough time to complete its operations without compromising the system's schedule. At the scheduler level, it's not permissible to include unconditional frames that are associated with either a sporadic or an event-triggered frame within the same schedule table as the sporadic or the event-triggered frame.

SIMPLICITY BELIES CHALLENGES

Although LIN is conceptually simple, device vendors still face significant challenges. The first difficulty is to fabricate bus transceivers that withstand automotive conditions, notably severe EMC-test compliance. Although LIN constrains interference generation through

baud- and slew-rate limiting, it's important that the system withstands severe levels of radiated and conducted emissions. Against a background of emissions and interference issues, car makers have developed a range of in-house tests for evaluating in-vehicle networks. These essentially consist of injecting RF interference into the bus and varying the signal's frequency, amplitude, and modulation depth until the system fails. Many common elements of these proprietary tests appear in the LIN-conformance test suite, which agencies such as the Communication and Systems Group at Fachhochschule University of Applied Sciences specialize in applying on behalf of its clients.

Scott Monroe, system architect at Texas Instruments' mixed-signal power and control group, explains that bulk-current-injection tests are popular in the United States, whereas European car makers favor DPI (direct-power-injection): "With DPI, you increase the RF-power levels into a bus of, say, three or four transceivers, via an RC coupler until the system breaks. With bulk current injection, the bus runs through a coupling coil, and you again vary the interference levels, looking for the point where message transmissions fail." Monroe notes that the LIN specifications don't mention protection against reverse-battery faults and negative-going transients, such as those that inductive loads create. These tests form part of the CISPR (International Special Committee on Radio Interface)-25 and ISO (International Organization for Standardization)-7637 transient-immunity standards for

automotive ICs. With a $\pm 40V$ bus fault and as much as 17-kV ESD protection, TI's TPIC1021 withstands these rigors and improves system reliability with features such as dominant-state time-out. Its I/O pins use a 5V-tolerant 3.3V structure for maximum logic compatibility. With thermal and bus-terminal protection from shorts to either supply rail, the chip doesn't disturb other bus communications in its inactive state. It responds to wake-up requests from the bus, from an enable pin that connects to the host microcontroller, or to a battery-voltage level-switch input. In sleep mode, quiescent current consumption falls from a maximum of around 2.5 mA to about 20 μA . The chip can also control an external voltage regulator, making it possible to power down a microcontroller or other LIN-protocol logic.

Other vendors that offer LIN transceivers include AMI Semiconductor, Atmel, Freescale, Infineon, Melexis, Microchip, On Semiconductor, Philips, STMicroelectronics, Yamar, and ZMD. Like the TI part, many of these devices offer similar pinout and functions to Freescale's MC33399 and Philips' TJA-1020 market-leading transceivers. Philips has a useful application note that illuminates LIN-transceiver issues (Reference 2). There are, however, detail differences between the electrical specifications in various competitive products, such as in the fault-voltage tolerance that vendors quote. For example, Atmel's ATA6661 withstands bus voltages of as much as 60V for use in 42V PowerNet environments. There are also some subtle differences

between apparently similar pinouts. For example, although most transceivers run directly from the vehicle's battery voltage, On Semiconductor's NCV7380/7382 requires a 5V supply on Pin 3, which is typically a battery-voltage-compatible wake-up-signal pin. The NCV7380 variant also dispenses with sleep-mode logic to minimize cost.

Gilles Guillaume of On Semiconductor's European marketing operation notes that the company offers designers extra flexibility, such as a voltage-regulator option to derive auxiliary power. In-

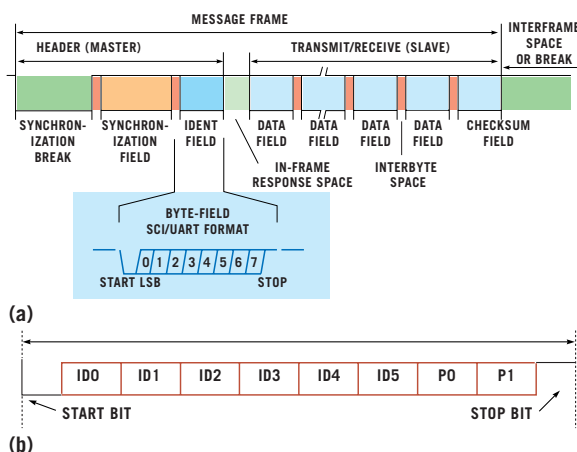


Figure 4 Each message exchange comprises a master-initiated header followed by a slave's reception or transmission (a). The master sends an identifier byte that uniquely defines the frame's purpose within a system (b).



tegrating a low-dropout-voltage regulator, the company's NCV7361A employs a modified eight-pin format to provide a 5V, 50-mA output. Melexis offers a similar part with its TH8061. Other examples of transceivers with integral voltage regulators include Microchip's MCP201, which trades the typical Pin 3 wake-up function to furnish a 5V, 50- μ A output. An external pass transistor can boost this capability for more demanding loads. An enhanced version, the MCP202, with higher ESD resistance and lower standby current will become available for sampling this summer. Texas Instruments also plans a transceiver variant with voltage-regulator-output capability.

MICROCONTROLLERS PLAY STATE MACHINES

The low cost and competitive nature of the LIN market complicate system integrators' choice of architecture. This choice even may differ within the same system due to the requirements of masters and slaves. Traditionally, pc-board designers gain maximum flexibility by using separate transceivers and microcontrollers. Because designers build these devices around a familiar product family and development-tool chain, this route is often a favorite, especially for masters. It may also provide an easy bridge into a CAN environment. Alternatively, microcontrollers with onboard transceivers and application-specific peripherals provide tightest integration, shrinking size and potentially offering the lowest cost for high-volume applications. Ross Mitchell, 8/16-bit-system software-application manager at Freescale, notes that size is often crucial, because trying to integrate, say, a mirror controller within plastic molding can

prove challenging: "It's typically desirable to have the control module as close as possible to its load, because this strategy minimizes wiring and can improve EMC performance."

But for lowest cost, state machines challenge microcontrollers and expensive on-chip memories. Such memories are necessary for supporting in-system configuration, which is one reason that some users want to stay with hardware that is compatible with LIN 1.3. Microchip's Johann Stelzer, European marketing manager for automotive products, believes that US customers in particular will adopt a subset of 2.0: "From a cost viewpoint, diagnostics and in-system configuration are a negative developments that threatens the \$1/node target," he says. More than one competitor acknowledges that today's low cost of CAN may erode the advantages of a complex LIN implementation. According to TI's Monroe, the optimal balance is crucially application-dependent. His company is supplying system makers with fully integrated products based on both state-machine and intelligent logic, although these devices are not yet available as catalog items. Many of these custom devices are three-pin slaves that fulfill roles as diverse as oil-quality and temperature sensors for engine-control systems to seat-weight sensors for occupant detection.

Monroe says, "From the lowest cost viewpoint, constraining feature creep and keeping it simple are obviously the ways to go." But in common with his peers, Monroe recognizes that there is a tendency among some designers to desire ever more features. As a result, a staggering array of controller options are

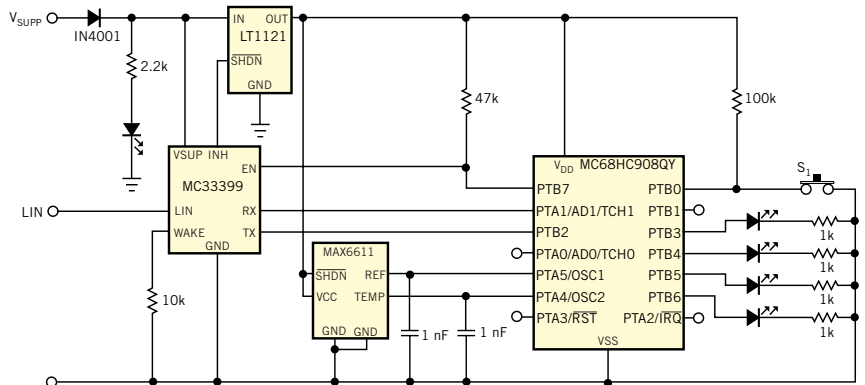


Figure 5 A reference design from Freescale illustrates a typical microcontroller-based implementation of a slave node.



available from vendors, including almost all of the transceiver suppliers. Currently, Atmel and On Semiconductor supply only the physical-layer interface, but both companies plan to offer the protocol logic, as well. Marcel Hennrich, Atmel's marketing manager for LIN products, confirms that derivatives of the company's AVR range will include the LIN transceiver, a 5V regulator, and a system watchdog. To assist its modular-development strategy, the company will offer a LIN-protocol stack from a third-party supplier to complement its normal tool chain. At On Semiconductor, the company's automotive-applications manager, Leo Airchriedler, says that the next step is to offer a LIN transceiver with an integral voltage regulator: "We will be carefully watching the market, and, if the demand is there, we will consider integrating the protocol logic, too." This combination forms the so-called SBC (system-basis-chip).

Available SBC examples include Freescale's MC33689, which bundles the transceiver, protocol handler, and voltage regulator into a 32-pin, 0.65-mm-pitch, surface-mount outline. This chip also includes three protected high-side drivers two of which support PWM; an uncommitted op amp intended as a current-sense amplifier; two high-voltage wake-up inputs; a configurable watchdog window timer; and an interrupt output that can signal undervoltage- and over-voltage-supply problems. An SPI port enables setup from a host, including several low-power-mode and slew-rate options. The chip also includes additional hardware-protection features, such as

overtemperature shutdown and over-current limiting.

Freescale shows some representative applications on its Web site, such as AN2623's temperature-sensor example (Figure 5). Here, the application runs under supervision from a 68HC908QY microcontroller with 4 kbytes of flash. The example shows the MC33399's handling the physical-layer-level translation and controlling the node's power supply by driving the Inhibit Pin of Linear Technology's LT1121 micropower low-dropout-voltage regulator. The software uses Freescale's LIN driver that is freely available from its Web site.

Freescale's Ross Mitchell says that it is just about possible to code a simple application into less memory, such as the 1.5 kbytes on the entry-level 908 family members, but considers 4 kbytes as a practical minimum. He and other vendors concur that the "sweet spot" for LIN lies somewhere around the 8-kbyte area for slaves, enabling applications such as window lifters with occupant-pinch detection. Intelligent-distributed-control chips, such as the MM908E625, employ the company's SmartMOS process to tighten integration for space-constrained nodes that require H-bridge motor control, such as headlamp levelers and mirror controllers. A new 908 family member, the MC68HC908QL4, includes an on-chip LIN-protocol handler and automatically synchronizes to the bus timing to suit slave use. The product is available now, and the company offers a \$199.95 evaluation board that complements its LINKit demo boards.

According to Microchip's Stelzer, the

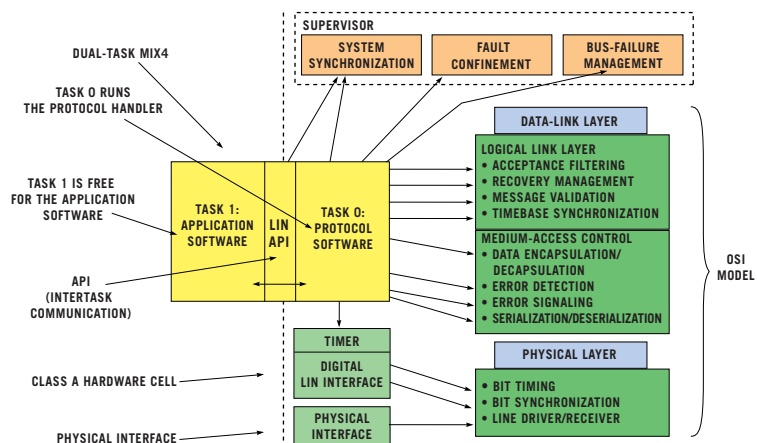


Figure 6 The MLX4 core from Melexis runs the LIN task and application software in two separate areas to emulate a two-task RTOS in hardware.



requirement for slaves to synchronize to the master's baud rate requires as many as 16 bits of timing resolution; hence, conventional UARTs aren't up to the job. For this reason, the company offers LIN-enhanced UARTs on chips such as its PIC16F688, which also includes features such as auto-wake-up on bus activity. This device uses the modified Harvard RISC architecture that is familiar to PIC users, and carries 4k words of program flash with 256 bytes of EEPROM for in-system-configuration data. Stelzer states that C programmers often favor higher end chips, such as the PIC18 family that provides more memory and substantially more computing power. The latest members are the PIC18F4x20/18F2x20 that offer 6 to 64 kbytes of program memory, speeds as high as 10 MIPS, and 28- and 44-pin QFN packages. They enjoy the same LIN enhancements as the F688, including Microchip's nanoWatt power-management technology that can reduce sleep-mode current drain to less-than-1- μ A levels. All Microchip's LIN-slave products include on-chip RC oscillators that eliminate external resonators or crystals. For master tasks, Stelzer recommends the PIC18F4680, a 64-kbyte device with 1 kbyte of data EEPROM and the company's LIN-specific enhancements among its total of 36 I/O pins. This chip also carries Microchip's enhanced CAN interface, enabling use as a CAN-to-LIN bridge.

Melexis took a different path with its LIN-specific MLX4 core. Michael Bender, the company's marketing manager for LIN products, says that this core appears in the TH8100, a single-chip slave for intelligent-switch modules, and forms the basis of several new chips that will roll out over the coming months. The TH8100 embodies the bus transceiver and its synchronization logic, with a LIN 1.3-compliant protocol handler and internal voltage regulation that powers the chip from the 12V rail. Requiring minimal external components, it includes 17 switch inputs, three ADC channels, and three PWM outputs. Its 4-bit MLX4 core has two independent register sets that partition and simultaneously handle the LIN protocol and the application (Figure 6). Each task owns a private set of peripherals, such as a timer and UART, and a private memory area. Flags and mutual-exclusion logic protect intertask data transfers to prevent both tasks from simultaneously writing to the same RAM address. Because register-

set switching occurs after every instruction, each task has 50% of the core's 4-MIPS capacity. The system can also dynamically share the available processing power; if one task enters wait mode, all processing power becomes available to the other task.

Additional sources of LIN-enabled hardware include Japanese microcontroller giants Fujitsu, NEC, and Renesas, as well as programmable-logic specialist such as Xilinx, and IP (intellectual-property)-core designers Intelliga and Fraunhofer Institut. LIN now appears on the 32-bit ARM platform, too, courtesy of Philips' SJA2020. Now available for sampling in a 144-pin package, this 60-MHz device carries 256 kbytes of flash and supports as many as six CAN channels and four LIN masters. Analog Devices intends to add a LIN 2.0-compliant transceiver to its ARM7-core AD μ C702x series, which currently supports the protocol via a UART-based software implementation.

This widespread support suggests a bright future for a technology that is only just beginning to emerge in production applications. Issues that industry insiders are keenly monitoring include the protocol's acceptance in the Japanese market, its progress toward ISO-standard recognition, and its penetration into areas outside automotive. As Microchip's Stelzer observes, freezing LIN 2.0 was an essential step toward getting the technology into design wins. He also notes that consumer applications, such as white goods, can be even cheaper by dispensing with the 12V medium: "They can use a single-wire, 5V system, because 12V is simply an extra effort." □

You can reach
Contributing Technical
Editor David Marsh
at forncett@
btinternet.com.

REFERENCES

1. Marsh, David: "Network protocols compete for highway supremacy," *EDN Europe*, June 2003, pg 26.
2. *TJA1020 LIN transceiver*, application note AN00093, Philips, 2002, www.semiconductors.philips.com.
3. For a list of vendors referenced in this article, please see the online version at www.edn.com.

TALK TO US

Post comments via TalkBack at the online version of this article at www.edn.com.

FOR MORE INFORMATION...

AMI Semiconductor

www.amis.com

Analog Devices

www.analog.com

Atmel

www.atmel.com

Bosch

www.bosch.de

**Fachhochschule
University of Applied
Sciences (Communi-
cation and Systems
Group)**

www.cs-group.de

Fraunhofer Institut

www.fraunhofer.de

Freescale

www.freescale.com

Fujitsu

www.fujitsu.com

IHR

www.ihr.de

Infineon

www.infineon.com

Intelliga

www.intelliga.co.uk

IXXAT

www.ixxat.de

LIN Consortium

www.lin-subbus.org

Linear Technology

www.linear.com

Melexis

www.melexis.com

Microchip

www.microchip.com

NEC

www.nec.com

On Semiconductor

www.onsemi.com

Philips Semiconductors

www.semiconductors.philips.com

Renesas

www.renesas.com

**Society of Automotive
Engineers (SAE)**

www.sae.org

STMicroelectronics

www.st.com

Texas Instruments

www.ti.com

Vector Informatik

www.vector-informatik.com

**Volcano Automotive
Group**

www.volcanoautomotive.com

**Warwick Control
Technologies**

www.warwickcontrol.com

Xilinx

www.xilinx.com

Yamar

www.yamar.com

ZMD

www.zmd.biz