

**THE EXPONENTIAL GROWTH OF ELECTRONIC SUBSYSTEMS WITHIN VEHICLES DEMANDS APPLICATION-SPECIFIC NETWORKING TOPOLOGIES THAT RANGE FROM THE MISSION-CRITICAL TO THE MUNDANE. EMBODYING EXPERIENCE FROM MORE COMPLEX PROTOCOLS, LIN TACKLES LOW-END APPLICATIONS WITH A COMPREHENSIVE DESIGN FLOW METHODOLOGY.**

# LIN simplifies and standardises in-vehicle networks

**W**ith 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 twenty years ago, the sole electronics content within the vast majority of vehicles was a radio. Then came the electronic ignitions, engine-management units and anti-lock-braking systems that are a standard fitment for today's entry-level cars. Now of course, better-specification vehicles include previously unheard-of luxuries, while top-of-the-range vehicles adopt sophisticated electronics, such as intelligent cruise controls that automatically maintain a safe distance to the driver ahead. Estimates suggest that the electronics content of an average car now accounts for no less than 22% of its build cost, creating new markets for the

embedded controllers, power devices, and communications technologies that link its electronic-control-units (ECUs).

Recognising the need for a robust in-vehicle network to manage distributed intelligence as well as

reducing wiring harness dimensions, Bosch designed CAN (controller-area-network) as long ago as 1986. Today, CAN dominates in-vehicle networking and has also made the transition to multiple industrial uses. Meantime, other networking systems have appeared to tackle emerging automotive applications. These include D2B (domestic-digital-data-bus), FlexRay, and MOST (media-oriented system transport), all of which employ fibre media for speed and EMC resistance. There are also time-triggered extensions to CAN (TT-CAN) that improve the protocol's determinism, as well as the TTP (time-triggered-protocol)

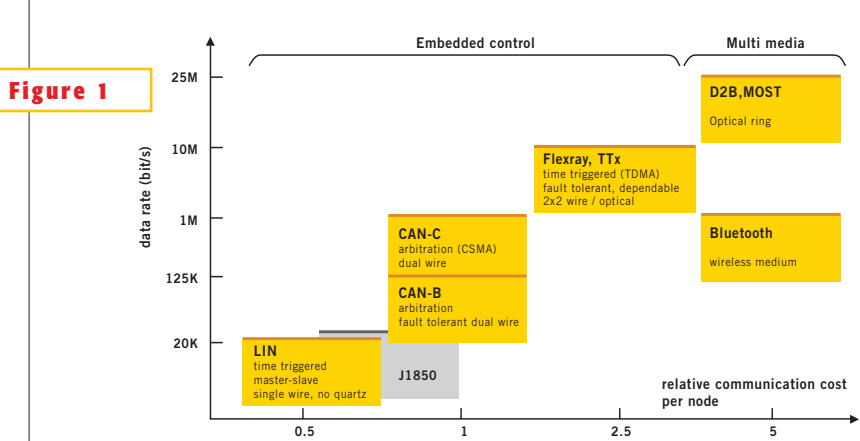
## AT A GLANCE

- ▷ LIN challenges CAN for lowest cost in non-critical applications
- ▷ The LIN specifications define its environment from physical-layer to toolchain
- ▷ Version 2.0 introduces new plug-&-play capability but costs memory
- ▷ Vendors offer a dazzling palette of hardware support
- ▷ Software tools enable automatic code generation
- ▷ Future growth prospects include consumer goods and industrial use

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 sun-roofs. As a result, carmakers increasingly embrace LIN local-area-interconnect), which positions itself at the lowest level in the automotive networking hierarchy (Figure 1).

The first LIN specification appeared back in 1999. Among the founder members of the LIN Consortium (its design authority) are carmakers BMW, Daimler-Chrysler, 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 carmakers use, as well as lessons accruing from many years of CAN evolution and development. Several amendments culminated in LIN version 1.3 of November 2002, which many observers regard as the first properly 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.

Meantime in North America, the Society of Automotive Engineers issued its J2602 recommended practice, "LIN Network for Vehicle Applications", with key carmaker 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



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

details, such as error handling. There is also the feeling among observers 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 just stop with low cost, although being markedly cheaper than CAN or the J1850 domestic US standard, is the prime driver. In fact, with original predictions for LIN lying 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 scaleable extension to CAN and J1850. It provides acceptable reliability for non-safety-critical tasks, with sub-100 msec response times and predictable worst-case timing characteristics. Learning from previous bus evolutions, the developers were also

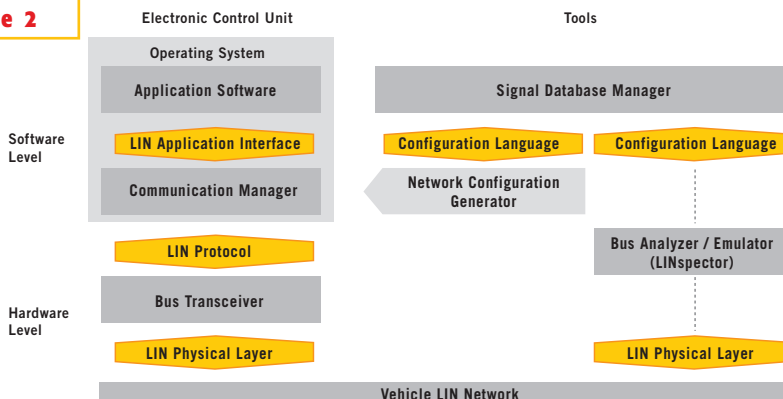
careful to consider toolchain support issues. Such considerations have become crucially important as the carmakers forge seamless co-development links with their systems suppliers.

Naturally, the specification must ensure hardware and software interoperability between multiple vendors, as well as minimising 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 application-programming interfaces (APIs) (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 multiple 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-&-play concept (see Sidebar "LIN 2.0 Goes Plug-&-Play").

**ONE-WIRE MASTER/SLAVE ARCHITECTURE CONSTRAINS COST**

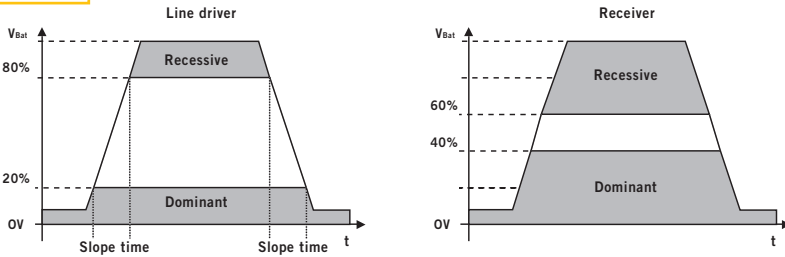
To minimise cost and wiring weight, LIN uses a single-conductor, wire-OR bus that takes advantage of the car's body shell to serve as a common ground. Each LIN

Figure 2



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

Figure 3



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

sub-net comprises one master and at least one slave node, up 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 CAN. Maximum transmission speed and reach is 20 kbps over 40m using UART/SCI communications. This makes LIN implementations possible 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-synchronisation facility that allows slave nodes to dispense with crystal or resonator timers.

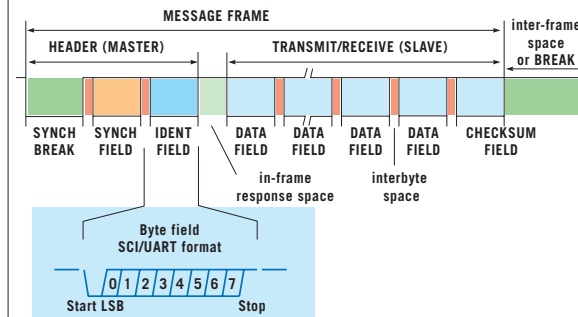
Physical-layer exchanges employ an enhancement to the ISO-9141 standard that dominates European and Japanese vehicle-diagnostic systems. LIN-compatible line drivers limit slew rate to around  $2\text{ V}/\mu\text{sec}$  to avoid creating interference due to fast edges. Such a line driver asserts the "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  $1\text{ k}\Omega$  resistor to battery voltage, and each slave defaults its I/O line high with a  $30\text{ k}\Omega$  pullup. 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  $220\text{ pF}$  to the line up to a maximum of  $10\text{ nF}$  per bus, which results in a system time constant of  $1 - 5\ \mu\text{sec}$ . In a region where AM radios still proliferate, SAE-J2602's

10.4 kbps limitation further eases compatibility issues for the North American market.

**MASTER/SLAVE ARCHITECTURE AVOIDS ARBITRATION**

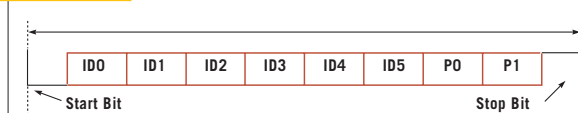
Unlike CAN, LIN's master/slave architecture avoids data-traffic collisions and the need for arbitration logic by having the master supervise message transmissions, thus ensuring that only one message transmits at any time. A frame consisting of the master's header, a pause, and a slave's response encapsulates each message exchange (Figure 4A). Start and stop bits surround each byte, resulting in a 10-bit transmission per byte. There are

Figure 4A



Each message exchange comprises a master-initiated header followed by a slave's reception or transmission

Figure 4B



The master sends an identifier byte that uniquely defines the frame's purpose within a system

several frame types, starting with diagnostic frames that carry diagnostic or initialisation information. By contrast, "unconditional frames" always carry signals of up to eight bytes. These are the frame types that typify applications. "Sporadic" frames also always carry signals, but slaves only respond if new data is available, otherwise leaving the data field blank—an attempt to add some dynamic behaviour into the system's schedule without compromising its determinism. To improve system responsiveness by reducing the bus traffic that polling infrequently-responding nodes generates, the protocol includes an event-triggered frame. This frame accommodates up to seven data bytes, as the first field carries an identifier that associates the frame with its task. Again, slaves only respond if they have new data. The protocol also provides 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 synch-break period, a single-byte synchronisation field, and an identifier byte. The latter 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 values between 0 - 59 (decimal) 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 synchronisation break and locks onto the synchronisation byte before scanning the bus message. One or more slaves then receives data, or a single slave transmits response data. The data field accommodates up to eight bytes, with its length predefined by association with its identifier byte. 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  $\mu$ sec – 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, as such behaviour indicates an error condition that would otherwise monopolise the bus. Slaves must wake up and be ready to

## LIN 2.0 GOES PLUG-&-PLAY

Until recently, the automotive industry was typically totally reliant on its systems suppliers for every aspect of electronics design. Unsurprisingly, this approach led to a plethora of single-source, special-to-type equipment that was expensive to develop, difficult to support, and next-to-impossible to reverse engineer. Other crucial issues include the massive expense of type approvals for safety-critical systems, which re-use over multiple vehicle platforms helps amortise. As a result, the industry has increasingly embraced the philosophy of co-operative development using standard toolsets.

This approach reduces costs and eases support issues, but still allows the carmaker to differentiate its products with elements of proprietary design. From the outset, LIN follows the open-systems-interconnect concept that hierarchically separates the networking hardware from application software (see main text). But to further stimulate interoperability and promote co-operative working practices, LIN includes a configuration language that enables software tools to configure individual nodes and the network as a whole. From a programmer's perspective, development starts

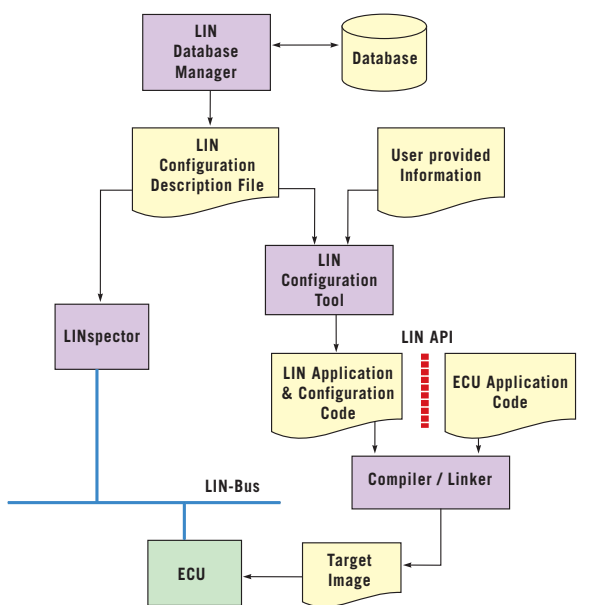
with defining all the nodes in the system and the relationships between them. Software tools, such as frame editors, compilers, and simulators, logically assign signals to nodes, pack them into frames, and run timing analyses for the composite system. Tools can also simulate missing nodes to facilitate virtual development. When simulation shows that the system meets its performance targets, the systems integrator can evaluate the design using real hardware. The general arrangement follows Volcano's original toolset, including its LinSpector bus analyser that has full emulation capabilities

language simplifies procuring and maintaining commonly-used components by standardising, for example, product identification codes. This capability language can also extend the reach of automation tools, such as code generators. In this instance, development starts with connecting each node in the new design to form a LIN cluster. The system definition tool parses each node and automatically creates the network's descriptor file. A system generator tool is then able to automatically generate initialisation code, effectively functioning as a plug-&-play tool.

(**Figure A**). Familiar from CAN and other in-vehicle networking topologies, additional tool suppliers include IHR, IXAT, Vector Informatik, Warwick Control Technologies, and Yamar. A key difference between LIN 1.3 and 2.0 is that the later specification mandates a number of node configuration commands. It's therefore essential for 2.0-compliant slave nodes to include rewritable, non-volatile memory. For this reason, 1.3-compliant slaves will run with a 2.0 master, as the later master gracefully avoids requesting the new features from older devices; conversely, 2.0 slaves won't run with 1.3 masters. There is also a new and optional diagnostic capability that uses a node-address-for-diagnostic (NAD) mechanism, where the master directly addresses a slave. A new LIN node capability

As the main text suggests, there's considerable debate among LIN watchers as to the merits of 2.0's enhancements. Chris Quigley, managing director at automotive networking tool-maker and design consultancy Warwick Control Technologies, acknowledges that some of his customers perceive that 2.0 compromises LIN's early simplicity. But he points out that the OEMs can reap benefits, such as product traceability, from following strictures such as the standard-supplier code philosophy. He also says that 2.0 fixes many of the vagaries of its predecessors, and considers the latest release "pretty mature". Overall, he assesses, "LIN's APIs, diagnostics, and product numbering features are more important than lowest possible cost-per-node, as they provide a complete, off-the-shelf solution."

**Figure A**



Volcano's suite is the original example of a complete LIN toolchain

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 timeslot. 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 will respond. 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 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 event-triggered frame.

radiated and conducted emissions. Against a background of emissions and interference issues, carmakers 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 & Systems Group at Fachhochschule University of Applied Sciences specialise in applying on behalf of its clients.

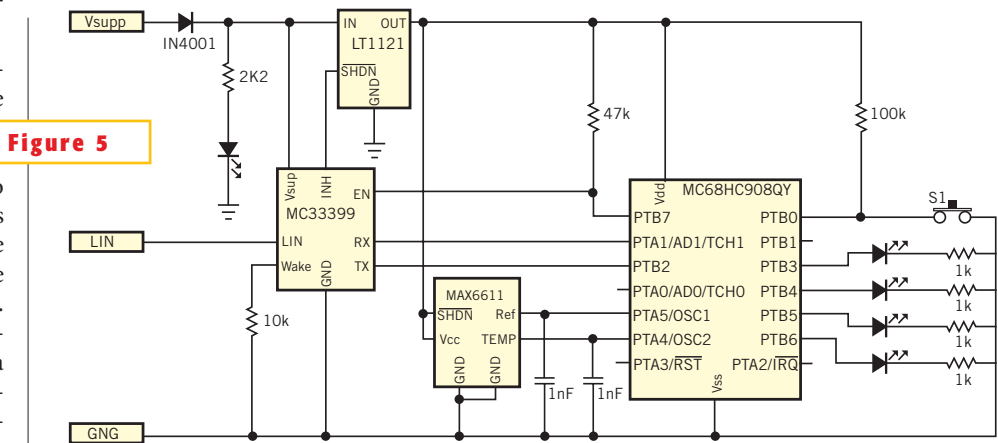
Scott Monroe, systems architect at Texas Instruments' mixed-signal power and control group, explains that bulk-current-injection tests are popular in the US, while European carmakers favour direct-power-injection (DPI): "With DPI, you increase the RF power levels into a bus of say, three or four transceivers, via an RC coupler up till the point that 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 specifically mention protection against reverse-battery faults and negative-going transients, such as inductive loads create. These tests form part of the CISPR-25 and ISO-7637 transient immunity standards for automotive ICs. With  $\pm 40V$  bus-fault and as much as 17 kV ESD protection, TI's TPIC1021 withstands these rigours and improves system reliability with features such as dominant-state timeout. Its I/O pins use a 5V-tolerant 3.3V structure for

maximum logic compatibility. With thermal protection and bus-terminal protection from shorts to either supply rail, the chip won'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  $\mu A$ . 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 Texas part, many of these devices offer similar pinout and functionality to Freescale's MC33399 and Philips' TJA1020 market-leading transceivers. (Philips has a very 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, while 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

**SIMPLICITY BELIES IMPLEMENTATION 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. While LIN constrains interference generation via baud- and slew-rate limiting, it's crucially important that the system withstands severe levels of



**Figure 5**

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



NCV7380 variant also dispenses with sleep mode logic to minimise cost.

Gilles Guillaume at ON Semiconductor's European marketing operation notes that the company offers designers extra flexibility, such as a voltage regulator option to derive auxiliary power. Integrating a low dropout voltage regulator, the company's NCV7361A employs a modified 8-pin format to provide a 5V, 50mA 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 µA output. An external pass transistor can boost this capability for more demanding loads. An enhanced version, the MCP202, will sample in the summer of 2005 with higher ESD resistance and lower standby current. 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 the system integrator's choice of architecture. This choice may even differ within the same system due to the different requirements of masters and slaves. Traditionally, circuit designers gain maximum flexibility by using separate transceivers and microcontrollers. Built around a familiar product family and development toolchain, this route is often a favourite, 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 applications manager at Freescale, notes that size is often crucial, as trying to integrate, say, a mirror controller within an existing plastic moulding can prove challenging: "It's typically very desirable to have the control module as close as possible to its load, as this strategy minimises wiring and can improve EMC performance."

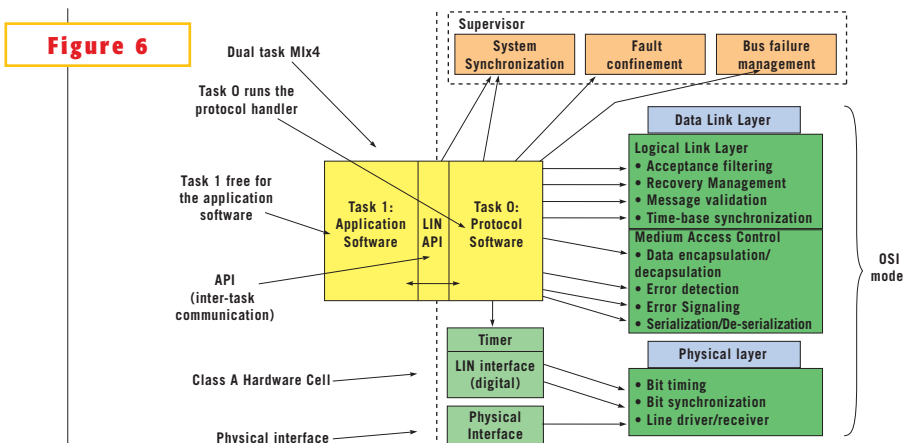
But for lowest cost, state machines challenge microcontrollers and expensive on-chip memories. Such memories are necessary to support in-system configuration, which is one reason why some users want to stay with hardware that is compatible with LIN 1.3. Microchip's Johann Stelzer, European marketing 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 development 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. Although not yet available as catalogue items, his company is supplying system makers with fully integrated solutions based on both state-machine and intelligent logic. Many of these custom devices are 3-pin slaves that fulfil roles as diverse as temperature and oil-quality 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 is obviously the way to go." But in common with his peers, Monroe recognises that there is a tendency among some designers to desire ever more features. As a result, there's a quite staggering array of controller options available from vendors including almost all of the transceiver suppliers. Currently, Atmel and ON Semiconductor only supply 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 toolchain. Over at ON Semiconductor, the company's automotive applications manager, Leo Airchriedler, says 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 system-basis-chip, or SBC.

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 under- and over-voltage supply problems. An SPI port enables set-up from a host, including several low-power modes and slew-rate options. The chip also includes additional hardware protection features, such as over-temperature shutdown and over-current limiting.

Freescale shows some representative applications on its website, such as AN2623's temperature sensor example (Figure 5). Here, the application runs under supervision from a 68HC908QY microcontroller with 4 kB of Flash. The example shows the MC33399 handling the physical-layer level translation and controlling the node's power supply by driving the Inhibit pin of Linear Technology's LT1121 micropower



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

low-dropout-voltage regulator. The software uses Freescale's LIN driver that is freely available from its website.

Freescale's Ross Mitchell says that it is just about possible to code a very simple application into less memory, such as the 1.5 kB on the entry-level 908 family members, but considers 4 kB as a practical minimum. He and other vendors concur that the "sweet spot" for LIN lies somewhere around the 8 kB area for slaves, enabling applications such as window lifters with occupant pinch detection. Intelligent-distributed-control (IDC) 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-levellers and mirror-controllers. A new 908 family member, the MC68HC908QL4, includes the LIN protocol handler on-chip and automatically synchronises to the bus timing to suit slave use. Available now, the company offers a \$199.95 evaluation board that complements its existing "LINKit" demo boards.

According to Microchip's Stelzer, the requirement for slaves to synchronise 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 together with 256 bytes of EEPROM for in-system configuration data. Stelzer states that C-programmers often favour 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 from 16 kB to 64 kB of program memory, speeds up to 10 MIPs, and tiny 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 sub-microamp levels. All Microchip's LIN slave solutions include an on-chip RC oscillator that eliminates external resonators or crystals. For master tasks, Stelzer recommends the PIC18F4680, a 64 kB device with 1 kB 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.

## FOR MORE INFORMATION...

**AMI Semiconductor**  
www.ami.com

**Analog Devices**  
www.analog.com

**Atmel**  
www.atmel.com

**Bosch**  
www.bosch.de

**Fachhochschule University of Applied Sciences (Communication & Systems Group)**  
www.cs-group.de

**Fraunhofer Institut**  
www.fraunhofer.de

**Freescale Semiconductor**  
www.freescale.com

**Fujitsu Microelectronics**  
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

Melexis took a different path with its LIN-specific MLX4 core. Michael Bender, the company's marketing manager for LIN products, says 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 synchronisation logic, together 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 MLX4 core is a four-bit machine with two independent register sets that partitions and simultaneously handles 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 protects intertask data transfers to prevent both tasks

You can reach  
Contributing Technical  
Editor David Marsh  
at [forncett@  
btinternet.com](mailto:forncett@btinternet.com)

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. It's also possible to 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 specialists such as Xilinx and IP-core designers Intelliga and Fraunhofer Institut. LIN now appears on the 32-bit ARM platform, too, courtesy of Philips' SJA2020. Sampling now in a 144-pin package, this 60 MHz device carries 256 kB of Flash, and supports up to six CAN channels and four LIN masters. Analog Devices intends to add a LIN 2.0-compliant transceiver to its ARM7-core AD $\mu$ 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 towards ISO standard recognition, and its penetration into areas outside of automotive. As Microchip's Stelzer observes, freezing LIN 2.0 was an essential step towards 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, as 12V is simply an extra effort." □

---

#### REFERENCES

1. Marsh, David: "Network protocols compete for highway supremacy", EDN Europe, June 2003, p.26. [www.ednmag.com](http://www.ednmag.com)
2. "TJA1020 LIN transceiver", application note AN00093, Philips 2002, [www.semiconductors.philips.com](http://www.semiconductors.philips.com)