



APPLICATION NOTE 3967

Selecting a Serial Bus

Abstract: This application note discusses the core of modern advanced electronics, which is typically a microcontroller that communicates with one or more peripheral devices that were originally memory mapped and connected to the data and address bus. However, due to high pin count, increased package size, and overall cost, another connection/communication scheme became the dominant choice—the serial interface. This article compares different serial interfaces, such as 1-Wire®, SPI, I²C, USB, etc., with respect to physical network size, network drivers, power supply, data rate, network inventory, and choice of functions. It shows how competitive and innovative the 1-Wire serial interface can be to the engineer trying to decide which serial interface to select.

This article was also featured in Maxim's [Engineering Journal](#), vol. 59 (PDF, 497kB).

The heart of today's advanced electronic products is a microcontroller (μC) that communicates with one or more peripheral devices. Initially, μC periphery was memory mapped and connected to the data and address bus. Chip-select signals were decoded from address lines to give each component its unique location in the limited address range. This type of interface determined the minimum number of pins (in addition to power and ground) to 8 (data) plus 1 (R/active-low W) plus 1 (active-low CS) plus n address lines [$n = \log_2(\text{number of internal registers or memory bytes})$]. A 16-byte device, for example, needed $8 + 1 + 1 + 4 = 14$ pins for communication. The access time was short, but the high pin count increased the package size and the overall cost. The obvious alternative to decrease cost and package size is a serial interface.

Defining a serial bus is not a trivial task. Besides data rate, bit sequence (most or least significant bit first), and voltage, one needs to consider the following:

- How the peripheral device gets selected (by hardware through the chip-select input or by a software protocol).
- How the peripheral device stays synchronized with the μC (through a hardware clock line or through clocking information embedded in the data stream).
- Whether data is transmitted on a single line (switching between "high" and "low") or a two-line differential connection (both lines changing their voltage simultaneously, but in the opposite direction).
- Whether both ends of the communication line are electrically terminated with a matching impedance (typical with differential signaling), or whether they are left unterminated or terminated at one end only (typical with single-ended buses).

Table 1 shows the matrix of variations using popular bus systems. Only 4 of the possible 16 combinations are currently known commercial products.

Beyond these parameters, the application can add further requirements such as the method of power delivery, isolation, noise immunity, the maximum distance between the μC (master) and the peripheral device (slave), or the structure of the cabling (linear, star, insensitive to reversal of wires). These requirements lead to applications such as building automation, industrial control, and reading of utility meters for which special standards have been developed.^{1, 2}

Table 1. Serial Bus Systems Overview

		ADDRESSING (SELECTION)			
SYNCHRONIZATION	PROTOCOL		CHIP-SELECT LINE		IMPEDANCE
SELF-CLOCKING	1-Wire [®] , LIN bus, SensorPath™				NOT MATCHED
		RS-485, LVDS, CAN, USB 2.0, FireWire [®]			MATCHED
CLOCK LINE					
	I ² C, SMBus™			SPI™, MICROWIRE™	NOT MATCHED
		SINGLE-ENDED	DIFFERENTIAL	SINGLE-ENDED	
TRANSMISSION MODE					

Requirements for Applications from Circuit Board to Backplane

A serial bus system for peripheral functions must not add any significant burden on the system. In particular,

- The connection must be easy to route (the fewer signals the better).
- The protocol must be easy to implement in software (or natively supported by the chosen $\mu\text{C}/\mu\text{P}$).
- There should be an adequate selection of device functions.
- The bus must be easy to expand.

The fewest number of signals is required with single-ended self-clocking systems that use a software protocol for addressing. As shown in Table 1, 1-Wire, LIN bus, and SensorPath meet these criteria. Within this group there are additional factors to consider (see **Table 2**).

Table 2. Further Distinctions Among 1-Wire, LIN Bus, and SensorPath Bus Systems

	1-Wire ³	LIN Bus ⁴	SensorPath ⁵
Physical Network Size	Board or backplane, can be expanded up to ~300m	~40m	Board
Network Drivers (Hardware)	Drivers are available for RS-232, I ² C, USB, and general μ P port pins ^{6, 7}	Drivers are available for μ P port pins	Super-I/O chips, μ P port pins
Network Drivers (Software)	Available free for various platforms, including μ Cs ⁸	Available free for Freescale™ μ Cs	Not available
Power Supply	Through the data line (typical case), local V _{CC} (some devices)	Through the data line	Local V _{CC}
Data Rate	Up to ~15kbps (standard) or ~125kbps (overdrive) ⁹	Up to ~20kbps	Data dependent, up to ~20kbps
Network Inventory	Through the "search ROM" network function	Not applicable, message-based addressing	Not supported
Choice of Device Functions	Large variety of device functions, including serial number, instrumentation, secure memories, etc.	Limited to functions needed in automotive applications	Limited to temperature sensors and voltage ADCs

Physical Network Size

Only SensorPath is limited to board-size applications. Under certain conditions and using appropriate hardware and software network drivers, the size of a 1-Wire bus can be expanded significantly.

Network Drivers

For protocol-based networks, one needs software drivers to generate the communication waveforms (link layer), to identify and address an individual slave/node in the network (network layer), and to transmit/receive data to or from a device (transport layer). Software drivers are specific for an operating system and communication port. There are 1-Wire hardware driver chips (masters) and adapters for ports such as COM, LPT, USB, and I²C. For large unterminated networks, reflections from cable ends, connectors, and stubs can limit the performance.

Power Supply

Each device in a network must be powered for operation. Most cost effective is a remote supply accessed through the data line. This method, also called "parasite power," makes it possible to read system diagnostic information (e.g., in power-down mode). For an example, refer to Figure 3 and the related text in application note 178, "[Printed Circuit Board Identification Using 1-Wire Products](#)."¹⁰ Parasite power, however, reduces the achievable data rate, as time must be set aside for power delivery.

Data Rate

Generally, a higher data rate is associated with a reduced network size and vice versa. In a 1-Wire system, due to the power-delivery feature, the maximum data rate depends on the number of slave devices in a network and the overall length (capacitance) of the cable.

Network Inventory

This feature allows the master to identify the number, type, and addresses of slave devices in a network. It is a prerequisite for networks with a dynamic (changing) population. Refer to page 22 of the [Dallas Engineering Journal](#) (vol. 2)¹¹ for an example.

Choice of Device Functions

The best interface is useless if the device functions required for an application are not available. Compared to LIN bus and SensorPath, 1-Wire has by far the largest selection of functions.

I²C/SMBus vs. 1-Wire

If the application can support a clock line, then the choice can be extended to I²C¹² and SMBus¹³ devices. In its initial specification, the SMBus was mostly a variation of the original 100kbps I²C bus specification with a timeout feature added. This timeout prevents the bus from becoming inoperable from a node that has lost synchronization with the bus driver; an I²C system requires a power-on reset to recover from this situation. In 1-Wire systems, the reset/presence-detect cycle resets the communication interface to a defined starting condition.

Besides the clock line, I²C/SMBus uses an acknowledge bit for every byte that is communicated on the bus. This reduces the net data rate by 12%. Transactions begin with a start condition followed by a device address and a data-direction bit (read or write), and they end with a stop condition. In 1-Wire systems, the requirements of the network layer must first be met (i.e., selecting a particular device or search ROM or broadcast); subsequent communication begins with a device-specific command code, which also affects the data direction (read or write).

A serious issue of the original I²C and SMBus is the limited 7-bit address space. With more than 127 different device types available, one cannot deduct the device function from its slave address. In addition, many I²C devices allow the user to set one or more address bits arbitrarily to have several equal devices on the bus. This feature also reduces the available address space. The standard method to work around address conflicts is splitting a bus into several segments and activating one segment at a time under software control. This segmentation requires more hardware and complicates the application firmware. The lack of an inventory function or enumeration makes it difficult for an I²C system to handle a dynamic population. This issue is solved with the Address Resolution Protocol of *SMBus Specification Version 2.0*.¹³ However, there are only a few SMBus devices available that support this feature.

SPI and MICROWIRE

SPI¹⁴ and MICROWIRE¹⁵, a subset of SPI, require an additional chip-select line for each device. Due to the chip-select signal, the SPI protocol only defines commands such as read and write for memory addresses and the status register. It does not use an acknowledge function. Typically, SPI devices have different pins for data input and data output. Because the data output is tristated (disabled) for anything except read functions, the two data pins can be tied together to form a single bidirectional data line. SPI is chosen for functions that are not available in other bus systems or for its fairly high data rate, which may be 2Mbps or higher. A downside of SPI and MICROWIRE is the decoder that generates the active-low CS signal to address an individual chip. However, there are no address conflicts. As with I²C, there is no inventory function. The master cannot deduct the device function from its logical address, which makes it difficult to manage a network with a dynamic population.

RS-485, LVDS, CAN, USB 2.0, and FireWire

These standards are discussed here as examples for differential signaling. The two fastest systems in this group, FireWire¹⁶ and USB 2.0¹⁷, are electrically point-to-point connections. Using sophisticated nodes or hubs, they

implement a virtual bus of a tree-like topology where data packets are transmitted from source to endpoint (USB) or peer-to-peer (FireWire) at a burst data rate of up to 480Mbps (USB 2.0) or 1600Mbps (FireWire). The limited packet size and the receive/buffer/retransmit communication concept add latency, which in turn reduces the achievable data throughput. Topology and protocol permit a maximum of 126 nodes for USB and 63 nodes for FireWire with a maximum distance of 4.5m between nodes using passive cable. Designed for applications such as PC periphery, multimedia, industrial control, and aviation (FireWire only), USB and FireWire devices can be plugged in without powering down the system (hot-swapped). This allows the network's population to change dynamically.

LVDS¹⁸, RS-485¹⁹, and CAN²⁰ implement a true linear bus structure with masters and slaves or even multiple masters. The fastest of these standards, low-voltage differential signaling (LVDS), can operate at 100Mbps if the bus size does not exceed 10m. Depending on the network size, the achievable data rate and throughput can be higher or lower. Designed as the electrical standard for backplane applications, LVDS allows hot-swapping, but does not specify any protocol.

RS-485 also specifies only the electrical parameters. Instead of nodes, RS-485 defines loads and the maximum number of loads per bus, which is 32. One electrical node can have a load of less than 1. Typical data rates are up to 35Mbps at 12m and 100kbps at 1200m network size, which are adequate for data acquisition and control applications. The protocol of RS-485 equipment is often based on components originally designed for RS-232.

The controller area network (CAN), in contrast, defines a serial communications protocol for distributed real-time control with a very high level of security, geared towards automotive applications and industrial automation. The data rate ranges from a network size of 1Mbps at 40m down to 50kbps at 1000m. The addressing is message-based without any protocol-inherent limitation to the number of nodes. CAN nodes can be hot-swapped for dynamic change of the network's population.

Summary

In the group of simple, low-cost bus systems, 1-Wire has the largest variety of device functions and network drivers as compared to LIN bus and SensorPath. I²C and SMBus require a clock line and V_{CC} power in addition to data and ground reference, but offer an impressive choice of device functions. SPI and MICROWIRE need an additional chip-select line, but communicate at significantly higher data rates. With its parasitic power supply and network inventory function, the 1-Wire interface and protocol support hot-swapping, a feature that is otherwise only found in high-speed systems that use differential signaling and in SMBus 2.0-compliant products. The best-known examples of hot-swapped 1-Wire devices are iButton® products, in which hot-swapping is the normal way of operation. 1-Wire devices have proven to be very efficient for functions such as global identification²¹, circuit boards/accessory identification and authentication¹⁰, temperature sensing, and actuation. Other very successful products are 1-Wire devices with secure memory and challenge-and-response functionality that protect intellectual property at a minimal cost through two-way authentication.^{22, 23}

A similar article appeared in the September 2006 issue of *Electronic Products*.

References

1. Interbus Club. www.interbusclub.com/ (industrial automation)
2. The valid M-Bus standard. www.m-bus.com/ (meter reading)
3. "Overview of 1-Wire Technology and Its Use." www.maxim-ic.com/AN1796 (intro to 1-Wire)
4. LIN Local Interconnect Network. www.lin-subbus.org/ (LIN specification)
5. National Semiconductor. "Cost Effective Partitioning of IO and Management Functions in PCs - Introduction of SensorPath™ Technology." www.national.com/nationaledge/jan04/article.html (SensorPath)
6. "Advanced 1-Wire Network Driver." www.maxim-ic.com/AN244 (hardware driver)
7. "Guidelines for Reliable 1-Wire Networks." www.maxim-ic.com/AN148 (1-Wire networks)
8. "1-Wire Software Resource Guide." www.maxim-ic.com/AN155 (software drivers)
9. "Determining the Recovery Time for Multiple-Slave 1-Wire Networks." www.maxim-ic.com/AN3829 (recovery time)
10. "Printed Circuit Board Identification Using 1-Wire Products." www.maxim-ic.com/AN178 (board

identification)

11. "Dallas Engineering Journal," vol. 2. pdfserv.maxim-ic.com/en/ej/DallasEJ2.pdf (dynamic network)
12. "The I²C-Bus Specification, Version 2.1, January 2000." www.nxp.com/acrobat_download/literature/9398/39340011.pdf (I²C)
13. "SMBus Specifications." www.smbus.org/specs/ (SMBus)
14. "M68HC11E Family." www.freescale.com/files/microcontrollers/doc/data_sheet/M68HC11E.pdf (SPI)
15. "MICROWIRE™ Serial Interface." www.national.com/an/AN/AN-452.pdf (MICROWIRE)
16. The Air Power Australia Website. "Firewire." www.ausairpower.net/OSR-0201.html (FireWire)
17. "USB 2.0 Specification." www.usb.org/developers/docs (USB specification)
18. National Semiconductor. "LVDS Owner's Manual: Low-Voltage Differential Signaling." www.national.com/appinfo/lvds/files/ownersmanual.pdf (LVDS)
19. Lammert Bies' Website. "RS485 serial information." www.lammertbies.nl/comm/info/RS-485.html (RS-485)
20. Robert Bosch GmbH. "CAN Specification, Version 2.0." www.semiconductors.bosch.de/pdf/can2spec.pdf (CAN specification part A)
CAN in Automation (CiA). "CAN Specification 2.0, Part B." www.can-cia.org/index.php?id=441 (CAN specification part B)
21. "Creating Global Identifiers Using 1-Wire Devices." www.maxim-ic.com/AN186 (global identifier)
22. "Protecting the R&D Investment—Two-Way Authentication and Secure Soft-Feature Settings." www.maxim-ic.com/AN3675 (two-way authentication)
23. "Xilinx® FPGA IFF Copy Protection with 1-Wire SHA-1 Secure Memories." www.maxim-ic.com/AN3826 (FPGA protection)

Also See

- [1-Wire and iButton](#)
- [Interface and Interconnect](#)
- [I²C: Need an I²C data converter? Maxim's got you covered!](#)
- [SMBus 2-Wire Interface Battery Management ICs](#)
- [USB Overview](#)
- [LVDS Line Drivers / Receivers](#)
- [Automotive Design Guide](#)
- [RS-485 Resources](#)

1-Wire is a registered trademark of Maxim Integrated Products, Inc.

FireWire is a registered trademark of Apple Computer, Inc.

MICROWIRE is a trademark of National Semiconductor Corp.

SensorPath is a trademark of National Semiconductor Corp.

SPI is a trademark of Motorola, Inc.

Application Note 3967: www.maxim-ic.com/an3967

More Information

For technical support: www.maxim-ic.com/support

For samples: www.maxim-ic.com/samples

Other questions and comments: www.maxim-ic.com/contact

Automatic Updates

Would you like to be automatically notified when new application notes are published in your areas of interest?

[Sign up for EE-Mail.](#)

Related Parts

ds1963s: [QuickView](#)

ds2401: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

AN3967, AN 3967, APP3967, Appnote3967, Appnote 3967

Copyright © by Maxim Integrated Products

Additional legal notices: www.maxim-ic.com/legal