ANLAN207

Management Interfaces for Micrel Switches

Introduction

Ethernet products typically need to be configured or managed either before or during operation. The configuration and management functions can take place either “in-band” or “out-of-band”. This application note provides information on five “out-of-band” interface options in more detail as well as providing suggestions for one “in-band” option that are available in the Micrel Ethernet products.

For example, a Micrel Ethernet switch product might interface to the fast-Ethernet MAC in a microprocessor using the media independent interface (MII) or reduced media independent interface (RMII) interface. Newer Ethernet switches also support Gigabit media independent interface (GMII) and reduced Gigabit media independent interface (RGMII) to connect to a 1G MAC. These interfaces are used only to transfer the data (Ethernet data packets), and not for writing and reading control and status registers. A separate interface is required to access the control and status registers of the switch and integrated PHYs for software control and switch management.

The “out-of-band” management interfaces are not only useful for managing and controlling the device, but can also be used for debugging. The entire register set can be read and dumped for viewing completely independent of any data transfers occurring.

The “in-band” management interfaces use the same path that is taken by the data and will have some impact on the throughput, depending upon the application.

The five “out-of-band” management interfaces that will be discussed in this application note are the following:

- MIIM (MDIO/MDC) being IEEE 802.3-compliant
- Serial Management Interface (SMI)
- Serial Peripheral Interface (SPI) (Slave Mode)
- I²C (Slave Mode)
- I²C (Master Mode – Serial EEPROM Interface)

Some specific Micrel switch products will be used as examples for the “out-of-band” management interfaces and tables will summarize the various features. Typical “in-band” management interfaces are generic host interfaces for 8/16 or 32-bit parallel busses, being described in more detail in Micrel’s application notes AN-137 and AN-140.

Another “in-band” management interface also not discussed here is the high-speed SPI being used for a tiny fast Ethernet controller (MAC/PHY) called KSZ8851SNL, which is described in the KSZ8851SNL datasheet.

MIIM – MII Management Interface for Ethernet PHYs

MIIM is an industry-standard serial management interface that is used to access the registers associated with PHY devices. It is defined in IEEE 802.3, Clause 22. The MIIM should not be confused with the MII interface which is used to interface a PHY device to a fast Ethernet MAC device for the purpose of transferring data packets. The MIIM is also known as the "MDIO/MDC Interface" and is typically supported by Ethernet PHY products industry wide.

The MIIM interface consists of two signals: MDIO (a bidirectional data line) and MDC (a clock line). 32 PHY registers can be accessed in 32 different PHY devices. The 32 PHY registers provide status information, control information, ID information, and manufacturer information for the addressed PHY device.

As MDIO normally is a (bidirectional) open-drain pin, an appropriate pull-up resistor (typically 1 kΩ – 4.7kΩ) is required on the MDIO signal line depending on the MDC clock rate and the number of devices attached to the MDIO line.

The MIIM frame format, which is based on 16-bit data, is shown in Table 1:

Table 1. MIIM Frame Format

Operation	Preamble	Start-of-Frame	Read/Write OP Code	PHY Address Bits[4:0]	REG Address Bits[4:0]	TA[1]	Data Bits[15:0][2]	Idle[3]
Read	32 1's	01	10	AAAAA	RRRRR	Z0	DDDDDDDD_DDDDDDDD	Z
Write	32 1's	01	10	AAAAA	RRRRR	10	DDDDDDDD_DDDDDDDD	Z

Notes:

1. TA = Turn-around bits.
2. x = "Don't care"; logic state can be "0" or "1" (as applicable).
3. Z = Tristate. The subsequent state "0" is driven by the slave device transmitting the read data starting one bit after.

MIIM Software Drivers

Many Ethernet software packages, like TCP/IP stacks, include an Ethernet PHY software driver which can be easily adapted for every Ethernet PHY that is IEEE 802.3 compliant. All Micrel PHYs support the MIIM standard, including older PHYs (KSZ8721, KSZ8001, KSZ8041, KSZ8021/31/51, and KSZ9021) as well as Micrel's latest fast-Ethernet PHY family, KSZ8081/91, and the Gigabit Ethernet PHY, KSZ9031.

The Micrel PHY software drivers are available for download from the Micrel website (www.micrel.com). Note that, in the source code of such PHY software drivers, the older Micrel PHYs are sometimes known under the manufacturer name "Kendin".

MIIM Interface for Switches with Embedded Ethernet PHYs

Most Ethernet switches have multiple integrated PHYs that support the MIIM, which allows reusing existing PHY software drivers written for the MIIM. However, as the switch is normally fully controlling the integrated PHYs on its own (like a stand-alone switch), no such PHY/switch software driver would be required. Using a PHY software driver with an Ethernet switch could result in unexpected behavior if it is not used properly. Micrel offers appropriate PHY software drivers for newer switch products. These switch software drivers are available for download from the Micrel website.

Serial Management Interface (SMI)

The serial management interface (SMI) interface is a Micrel proprietary interface that is used to read and write registers in the device. It uses the same MDIO/MDC pins as the MIIM, but applies different protocols on those pins. The protocol and format of the data will be dependent on which specific Micrel device is being accessed.

Table 2 and Table 3 show the SMI frame format for the KSZ8864/95 and for the KSZ8863/73/93 products which is based on 8-bit data, but still transmitted in the 16-bit frame format of MIIM.

Table 2. SMI Frame Format (KSZ8864/95 products)

Operation	Preamble	Start-of-Frame	Read/Write OP Code	PHY Address Bits[4:0]	REG Address Bits[4:0]	TA[1]	Data Bits[15:0][2]	Idle[3]
Read	32 1's	01	10	RR11R	RRRRR	Z0	0000_0000_DDDD_DDDD	Z
Write	32 1's	01	01	RR11R	RRRRR	10	xxxx_xxxx_DDDD_DDDD	Z

Table 3. SMI Fame Format (KSZ8863/73/93 products)

Operation	Preamble	Start-of-Frame	Read/Write OP Code	PHY Address Bits[4:0]	REG Address Bits[4:0]	TA[1]	Data Bits[15:0][2]	Idle[3]
Read	32 1's	01	00	1xRRR	RRRRR	Z0	0000_0000_DDDD_DDDD	Z
Write	32 1's	01	00	0xRRR	RRRRR	10	xxxx_xxxx_DDDD_DDDD	Z

The read/write OP code is the same for MIIM and SMI for KSZ8864/95, except that the upper byte of the 16 data bits is 0000_0000 for read and "don't care" for write. Also, the PHY address is encoded in the 8-bit register address, whereas for the KSZ8863/73/93 devices, the read/write OP code = 00 + PHY Address Bit[4] is different. In addition, the 8-bit register address is encoded differently for these switches.

An appropriate pull-up resistor (typically 1kΩ to 4.7kΩ) is required on the MDIO signal line depending on the MDC clock rate and the number of devices attached to the line.

For an SMI read command, MDIO must be switched to input on the MAC side during the Z state of TA. SMI will respond starting with "0" of TA (Table 2 and Table 3).

The newer switches KSZ8765/75 and KSZ8794/95 do not support SMI, only MIIM or SPI.

Differentiating Points between SMI and MIIM

Note that standard MACs of microcontrollers do not directly support SMI on their MDIO/MDC pins, only the MIIM. Therefore the SMI must be emulated by microcontroller software. This is known as “bit banging”. If, however, the MAC function is implemented using IP in an FPGA, for example, a standard MIIM could be modified to support SMI as well.

Pull-Up Resistors on the SMI and MIIM Management Busses

The SMI and MIIM management busses allow multiple devices to be interconnected. To prevent any collision of the data outputs, except for SPIQ (the SPI data output) these outputs are open-drain outputs and therefore appropriate pull-up resistors are required. These are typically connected to the digital I/O supply voltage for flexibility and to match with the I/O supply voltage of the attached Ethernet MAC, which for all newer devices can be chosen from 1.8V, 2.5V, and 3.3V.

A typical value for these pull-up resistors is in the range of 1kΩ to 4.7kΩ and depends on the number of pins being interconnected, the data rate at these pins, and the digital I/O supply voltage.

It is advisable to check the slew rate of the open-drain signals to make sure that the data can be supported properly. Higher speed requires lower pull-up resistor values. However, all attached output pins must be able to sink the resulting current and still provide the required logic LOW level. If data cannot be transferred properly, check the signals by using an analog or high-bandwidth digital scope, instead of a logic analyzer, to see whether the clock and data edges are fast enough.

Serial Peripheral Interface (SPI)

The SPI slave interface is implemented on many of Micrel's switch products. It is an industry-standard, four-line serial interface.

The signal usage is as follows:

- Chip select in SPI slave mode, also called SPIS_N (Micrel nomenclature) or SPISN or /SS
- SPI clock input in slave mode, also called SPIC (Micrel nomenclature) or SCL or SCLK
- SPI data in in slave mode, also called SPID (Micrel nomenclature) or SDA or MOSI
- SPI data out in slave mode, also called SPIQ (Micrel nomenclature) or MISO

Normally, SPI master modes 0 or 3 are used by the attached microcontroller where the falling SCK edge shifts the data and the rising SCK edge captures the data. Additional details are available in the device datasheets.

Table 4 provides a summary of the management interfaces being supported for the various devices and Table 5 provides a summary of the maximum clock rates. Only KSZ8893 and KSZ8864/95 allow using two of the supported management interfaces at the same time. For the other devices being listed only one of the given management interfaces can be used at the same time.

Table 4. Summary of Management Interfaces Supported Specific Devices

Device	MIIM	SMI	SPI	I^2C (Slave)	EEPROM I^2C (Master)	Notes
KSZ8463	X		X			"Either/or", normally SPI.
KSZ8765, KSZ8775, KSZ8794, KSZ8795	X		X[4]			"Either/or", normally SPI.
KSZ8863, KSZ8873	X	X	X	X	X	Only one out of five. EEPROM excludes switch management.
KSZ8893	X	X	X	X	X	Multiple management interfaces are possible at the same time.
KSZ8864, KSZ8895	X	X	X		X	Multiple management interfaces are possible at the same time.
KSZ8441, KSZ8462, KSZ8852					X (Host MAC Address)	Management "in-band" through host interface.

Note:

4. For the KSZ87xx switch devices, the SPI frame format is different from other devices. The standard SPI frame starts with one byte command + one byte address + data bytes. For KSZ8765/75 and KSZ8794/95, a 12-bit address (still 8 lower bits are used only) must be shifted left by one bit location (new LSB can be either 1 or 0) and is combined with the 3-bit read/write command and then is sent as 2x 8 bytes. The rest of the SPI frame stays the same as usual.

Table 5. Summary of Maximum Clock Rates for Management Interfaces

Device	f_{MIIM} (MHz)	f_{SMI} (MHz)	f_{SPI} (MHz)	f_{I2C} (MHz)	Notes
KSZ8463	5	–	50	–	Either MIIM or SPI is available. For IEEE 1588v2 applications, SPI is required.
KSZ8765, KSZ8775, KSZ8794, KSZ8795	25	–	50	–	Either MIIM or SPI is available. The SPI clock edge is programmable. Default is the falling clock edge.
KSZ8863/73	5	5	25	2.5	Only one out of five options is available at the same time (selected by pin strapping). The SPI clock rate range can be selected in switch Register 11, Bits[7 – 6]. The default SPI clock setting is for ≥12.5MHz.
KSZ8864, KSZ8895	10	10	25	–	Flexible options. The SPI clock rate range can be selected in switch Register 11, Bits[5 – 4]. The default SPI clock setting is for ≥12.5MHz.
KSZ8893	5	5	5	2.5	Flexible options.
KSZ8851SNL	–	–	40	–	"In-band" management (controller).

I^2C Slave Mode

This slower method of accessing the device registers is currently offered on older Micrel switch devices KSZ8893 and KSZ8863/73. To properly select the I^2C slave mode by pin strapping, see the I2C Master Mode – Serial EEPROM Interface section as well as Table 4 and Table 5).

I^{2}C Master Mode – Serial EEPROM Interface

Many Micrel switch products offer the ability to load the device registers from content stored on a serial EEPROM. The load operation takes place after the power-on reset (POR) function has completed and a specified short stabilization time has elapsed. It provides the ability to configure the device automatically without firmware and operate the switch in what is called “unmanaged mode”.

If the strapping options offered by a device are not sufficient and no microcontroller is available to configure the device via software, the serial EEPROM can be used as a vehicle to automatically program certain device registers after POR.

It is important to note that the various Micrel products that incorporate this feature require a specific type of serial EEPROM which may be different for each product. Be sure to consult the specific device datasheet for which serial EEPROM is required.

Additionally, the pins that are used for the serial EEPROM interface may be multiplexed with other functions, so exact usage and setup may differ from device to device.

The number and range of registers that are automatically read by the device into the internal register space will vary per device. Table 6 provides some examples of this for several devices. Only the writable registers (not the read only or status registers) will be written to using contents from the serial EEPROM.

Typically, if the first byte or word does not match what the device is expecting, the remainder of the Serial EEPROM read operation will not take place and the registers will remain in their default values. Refer to the specific device datasheet.

Various Micrel products also allow the writing (or reading) of the serial EEPROM contents using specific device registers. Refer to the specific device datasheet.

Table 6. Micrel Device Serial EEPROM Contents

Device	Serial EEPROM Contents Transferred
KSZS8462, KSZ8441, KSZ8852	16-bit words from 0h to 06h are read from the serial EEPROM. These specific words must be the host MAC address.
KSZ8863, KSZ8873, KSZ8893	199 registers (0 – 198) are loaded from the serial EEPROM.
KSZ8864, KSZ8895	256 registers (0 – 255) are loaded from the serial EEPROM.

Programming the Serial EEPROM

There are two options for programming the serial EEPROM:

1. Program the EEPROM externally and put it in a socket or solder the EEPROM after programming.
2. Program the EEPROM on the PCB using an in-system microcontroller.

The first option is more labor intensive and requires a socket or extra handling during production. For the not-so-common second option, the attached management microcontroller could program (or reprogram) the EEPROM using I²C while keeping the attached switch in I²C slave mode (in the case of the KSZ8863/73, e.g.) or in reset mode while programming the EEPROM and releasing the switch reset line in EEPROM/I²C master mode after programming to finally load the programmed settings automatically from the EEPROM into the switch – by the switch, not by the microcontroller.

For the KSZ8863/73 devices, switch management is not possible in EEPROM/I²C master mode, whereas for the KSZ8864 and KSZ8895 devices the interface mode still can be changed according to Table 7 while the switch is already running.

Table 7. Serial Bus Strapping Selection (KSZ8864 and KSZ8895 devices)

PS[1:0]	Serial Bus Configuration
00	I^{2}C Master Mode for EEPROM
01	SMI Interface Mode
10	SPI Slave Mode for CPU Interface
11	Factory Test mode (BIST)

In I²C master mode, the KSZ8864/95 and KSZ8863/73 (also older KSZ8893) switches use the device address 1010_000x (x = 1 for read command and x = 0 for write command) for the attached EEPROM, whereas in I²C slave mode, the device address for the KSZ8863/73 and KSZ8893 switches is 1011_111x (again x = 1 for read command and x = 0 for write command). The KSZ8864/95 switches support SPI/SMI/MIIM/I2C master mode and newer KSZ8765/75 and KSZ8794/95 only support SPI/MIIM and not any EEPROM I^{2}C master mode.

Micrel's evaluation boards allow the user to program/reprogram the strapping EEPROM and to access all switch registers via a USB interface by Windows®-based software (called “Utilities”), which can be downloaded from the Micrel website.

Selecting and Using Multiple Management Interfaces on One Device

Some Micrel devices offer more than one management interface option. In those cases, the choice of which one(s) to use will be based on the following:

- Using the device in a design that uses a specific management interface.
- Requirement to use un-managed mode.
- Requirement to access all registers or only PHY registers in a switch.
- Ability to do debugging.

The following sub-sections provide some real device examples of multiple management interfaces in one device and some things that the user needs to be aware of.

Management Interfaces – KSZ8863 and KSZ8873 Devices

While Table 7 details the management interfaces that are configurable via the strapping pins on the KSZ8864 and KSZ8895 devices, Figure 1 illustrates the management interfaces available in the KSZ8863 and KSZ8873 devices. These families of switch devices incorporate all five of the different interfaces.

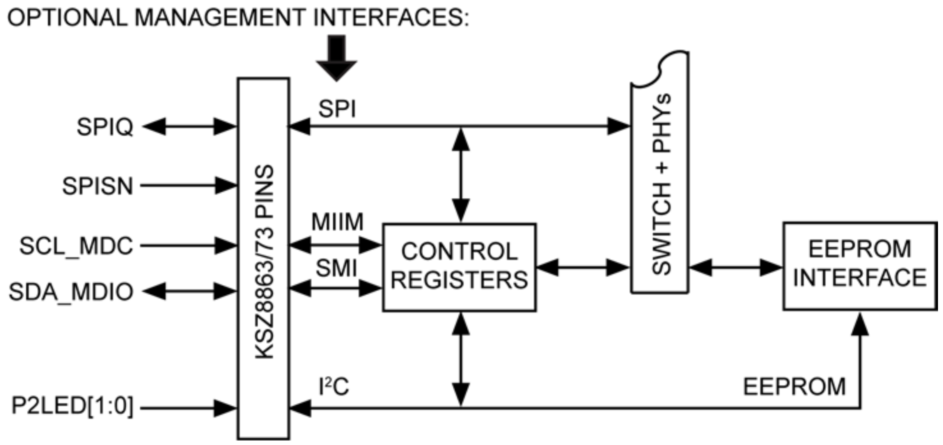

Figure 1. Management Interfaces for the KSZ8863/73 Devices

Table 8 provides a summary of which management interface is available for the KSZ8863/73 devices, how it is selected, and which pins are used for which interface signal/function.

Table 8. Selection of Management Interfaces for the KSZ8863/73 Devices

P2LED[1:0]	Serial Bus Configuration	Device Pins			
		SCL_MDC	SDA_MDIO	SPIQ	SPISN
00	I²C Master Mode for EEPROM	I²C Clock	I²C Data I/O	Not Used (Tri-Stated)	Pull Externally HIGH
01	I²C Slave Mode	I²C Clock	I²C Data I/O	Not Used (Tri-Stated)	Pull Externally HIGH
10	SPI Slave Mode	SPI Clock	SPI Data In	SPI Data Out	SPI Chip Select
11 (Default)	SMI/MIIM Mode	SMI/MIIM Clock	SMI/MIIM Data I/O	Not Used (Tri-Stated)	Pull Externally HIGH

The interface mode is selected when the P2LED[1:0] pins (strapping pins) are sampled by the device at the end of the POR time (rising edge of the RSTN signal). Each of these four pins will need the appropriate pull-up or pull-down resistor installed so that logic 0 or 1 is captured on each pin. The four modes are explained for KSZ8873/73 as follows:

- **P2LED[1:0] = 00:** This puts the device in the mode where, at the end of the POR cycle, a pre-determined amount of bytes/words will be read from the serial EEPROM and loaded into specific device registers. Two of the four device pins are dynamically used and one of the four pins is statically used. Refer to the specific device datasheet for details.
- **P2LED[1:0] = 01:** This puts the device in the mode where, at the end of the POR cycle, the device will be expecting to communicate via the I^{2}C interface. Two of the four device pins are dynamically used and one of the four pins is statically used. Refer to the specific device datasheet for details.
- **P2LED[1:0] = 10:** This puts the device in the mode where, at the end of the POR cycle, the device will be expecting to communicate via the SPI Interface. All four device pins are dynamically used. Refer to the specific device datasheet for details.
- **P2LED[1:0] = 11:** This this default setting puts the device in the mode where, at the end of the POR cycle, the device will be expecting to communicate via the SMI or MIIM Interfaces. The devices will be able to automatically differentiate which method is used and adjust to communicate using that interface.

Management Interfaces – KSZ8463 Device

Note that for the KSZ8463 switch device, either the MIIM or the SPI Interface can be chosen via pin strapping. However, to support IEEE 1588v2 and the corresponding PTP software stack, the SPI management interface must be selected. The MIIM interface option cannot be used for this application because it does not provide access to the entire switch registers.

Port Numbering

If an existing PHY software driver (typically set up for the default PHY address 1) is used with a 3-port switch having two PHY ports called PHY1 (using PHY address #1) and PHY2 (using PHY Address 2) in addition to the MII/RMII (which then is Port 3), traffic would be prevented from flowing between PHY2 and Port 3 (the microcontroller, e.g.) if no link is available at PHY1.

As a result, such a PHY software driver being used for an Ethernet switch would need to be extended for the number of PHY ports (and their corresponding PHY addresses) being used. Otherwise, a switch PHY port having no link could prevent Ethernet traffic between other ports, including the MII/RMII, where the MAC of a microcontroller is connected.

Typically, a switch does not need a PHY software driver. It is only required if the controlling system must know the status of the various PHY ports. For the overall switch management, a more powerful management interface like SPI is used, allowing access by the attached microcontroller to all switch registers, not just the PHY register sets (one register set per PHY port) being covered by the MIIM.

The PHY ports of the switches need to be addressed properly if any PHY software driver is used.

The default setting for KSZ8895 is PHY address 1 for PHY Port 1, PHY Address 2 for PHY Port 2 and so forth up to PHY Address 5 for Port 5, whereas for the KSZ8864 device, PHY Address 2 is for PHY Port 1 and PHY Address 3 is for PHY Port 2.

If multiple switches are managed by the same MIIM or SMI, the PHY addresses must be different for all PHY ports, as usual. For example, to manage multiple switches using the KSZ8863/73 (and KSZ8893), the PHY addresses must be changed (at least for the additional switches) in MIIM Register 15 (0x0f), Global Control 13.

Note that the KSZ8864/95 switches and newer KSZ8765/75 and KSZ8794/95 do not allow changing the PHY Port Addresses 1 up to 5.

Since the I²C device addresses are fixed in I²C slave mode, no multiple switches can be managed in I²C slave mode by using the same I²C bus. Compare this to SPI slave mode, where multiple switches can be managed by the same SPI, which then requires separate enable lines for every switch. Note that SPI daisy chaining is not supported.

PHYs always offer a few strapping pins to define the PHY address. But be aware that PHY address zero can be a broadcast address which is handled differently.

Management Mode and Strapping Pins

Depending on the specific device, the strapping pins will not only select the desired management interface but will also select other functions.

The strapping pins, in most cases, incorporate internal pull-down or pull-up resistors. However, it is advisable to use external strapping resistors (typically 1kΩ for pull-down and 4.7kΩ for pull-up resistors) as the internal pull-down/up resistors have relatively high resistance values of 40kΩ to 58kΩ (depending on the device) and are not very accurate (±30%).

If PCB space or additional cost is critical, a few less critical strapping resistors could be omitted. Then the actual setup should be checked by software via management interface to make sure that the wanted features are selected properly after POR has concluded. If not, it needs to be corrected via the management interface. When deciding to use or not use the external resistors, be advised that CMOS input pins should not be floating.

In any case, external resistor strapping is required if the desired feature needs a strapping polarity opposite to the default setting – if not done by any other method.

The lowest priority for strapping is the default register setting, which is described in the device datasheets. The default setting can be overwritten by external resistor strapping, especially if the logic level is opposite to the default setting. Alternately the switch setup can be managed through one of the management interfaces by an attached microcontroller, which is highest priority.

Note: Besides the strapping function (which then is a CMOS input to the device) many strapping pins are used as CMOS push/pull outputs (like status LED outputs, e.g.) immediately after the pin strapping is read. To prevent that the pin strapping is affected, don’t connect any low impedance bipolar circuits (including bipolar transistors and their base resistor) to such pins, but use high impedance CMOS buffers etc., instead.

Starting the Switch Automatically

Depending on the switch device, the switch will start automatically after POR or will not.

Both KSZ8864 and KSZ8895 are starting automatically after POR only if the EEPROM master mode is selected, independent whether a strapping EEPROM is connected (non-programmed or having a valid content in the first byte) or not.

If a strapping EEPROM should be used, besides having a valid data in the first byte, also the "start switch" bit must be set in the relevant switch register location.

It must be noted, that KSZ8864/95 (like older switches KSZ8893 and KSZ8995) do not start automatically if any of the management modes are selected – even though the PHY ports of the switch still can link up with attached link partners! That means these switches in such a case always must be started by the management microcontroller to make the switching working.

The KSZ8864/95 (and the older KSZ8893 and KSZ8995) allows change to the configuration mode on the fly, for example to configure and/or start the switch automatically after power-on without using any microcontroller (especially if multiple switches are connected in a daisy chain and data transmission through the switches should be able immediately), but still allow switch management afterwards by selecting any of the supported management interfaces according to Table 4.

In contrast to KSZ8864/95, the switches KSZ8863 and KSZ8873 always start automatically after power-on and the management interface option cannot be changed on the fly. Therefore, if configuration or management by a microcontroller is required, these switches must be configured in any of the management interface modes, and not in EEPROM master mode (Table 5).

Summary

This application note has briefly surveyed the management interfaces that are implemented in various Micrel Ethernet switches and controllers. While there is much commonality, we have pointed out the important differences between interfaces protocols and between devices. The reader should now have a good understanding of the issues and relative merits of these interfaces, but the individual device datasheet must always be referenced as the definitive source for the important details.

MICREL, INC.   2180 FORTUNE DRIVE   SAN JOSE, CA 95131   USA
TEL +1 (408) 944-0800 FAX +1 (408) 474-1000 WEB http://www.micrel.com

Micrel makes no representations or warranties with respect to the accuracy or completeness of the information furnished in this data sheet. This information is not intended as a warranty and Micrel does not assume responsibility for its use. Micrel reserves the right to change circuitry, specifications and descriptions at any time without notice. No license, whether express, implied, arising by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Micrel's terms and conditions of sale for such products, Micrel assumes no liability whatsoever, and Micrel disclaims any express or implied warranty relating to the sale and/or use of Micrel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right.

Micrel Products are not designed or authorized for use as components in life support appliances, devices or systems where malfunction of a product can reasonably be expected to result in personal injury. Life support devices or systems are devices or systems that (a) are intended for surgical implant into the body or (b) support or sustain life, and whose failure to perform can be reasonably expected to result in a significant injury to the user. A Purchaser's use or sale of Micrel Products for use in life support appliances, devices or systems is a Purchaser's own risk and Purchaser agrees to fully indemnify Micrel for any damages resulting from such use or sale.