Freescale Semiconductor
White Paper

Document Number: PQIIPROWP
Rev. 0, 08/2010

PowerQUICC II Pro for Entry-Level Industrial Networking and Control Solutions

by *Networking and Multimedia Group*
Freescale Semiconductor, Inc.
Austin, TX

Freescale's PowerQUICC II Pro processor family has long established itself as the premier, cost-effective communications processor family in the market. These processors are widely used in networking, I/O control processing, and intelligent NIC applications. The MPC830*x* family extends current PowerQUICC offerings into entry-level, industrial applications by improving the price-to-performance ratio, increasing industrial functionality, and enhancing interface efficiencies. The MPC830*x* family consists of the MPC8306, MPC8306S, MPC8308, and MPC8309. This white paper focuses on the usage of the MPC8306 and MPC8309 in factory automation/industrial control and other low-end embedded applications.

Unless otherwise specified, MPC830*x* applies to all devices in the family.

Contents

1. Product Features 2
2. Industrial Networking and Control Solutions Based on PowerQUICC 9
3. Use Cases/Applications 11
4. Conclusion 24
5. Revision History 25

freescale semiconductor

1 Product Features

The MPC830*x* devices are part of the PowerQUICC II Pro architecture designed to meet the cost and performance requirements of the entry-level networking and industrial space. Based on the highly successful PowerQUICC system-on-chip (SoC) platform, which has been well proven in traditional networking, these new devices extend this platform by offering additional integration and application content processing capabilities at a lower cost entry point.

Figure 1 shows the MPC8309 block diagram.

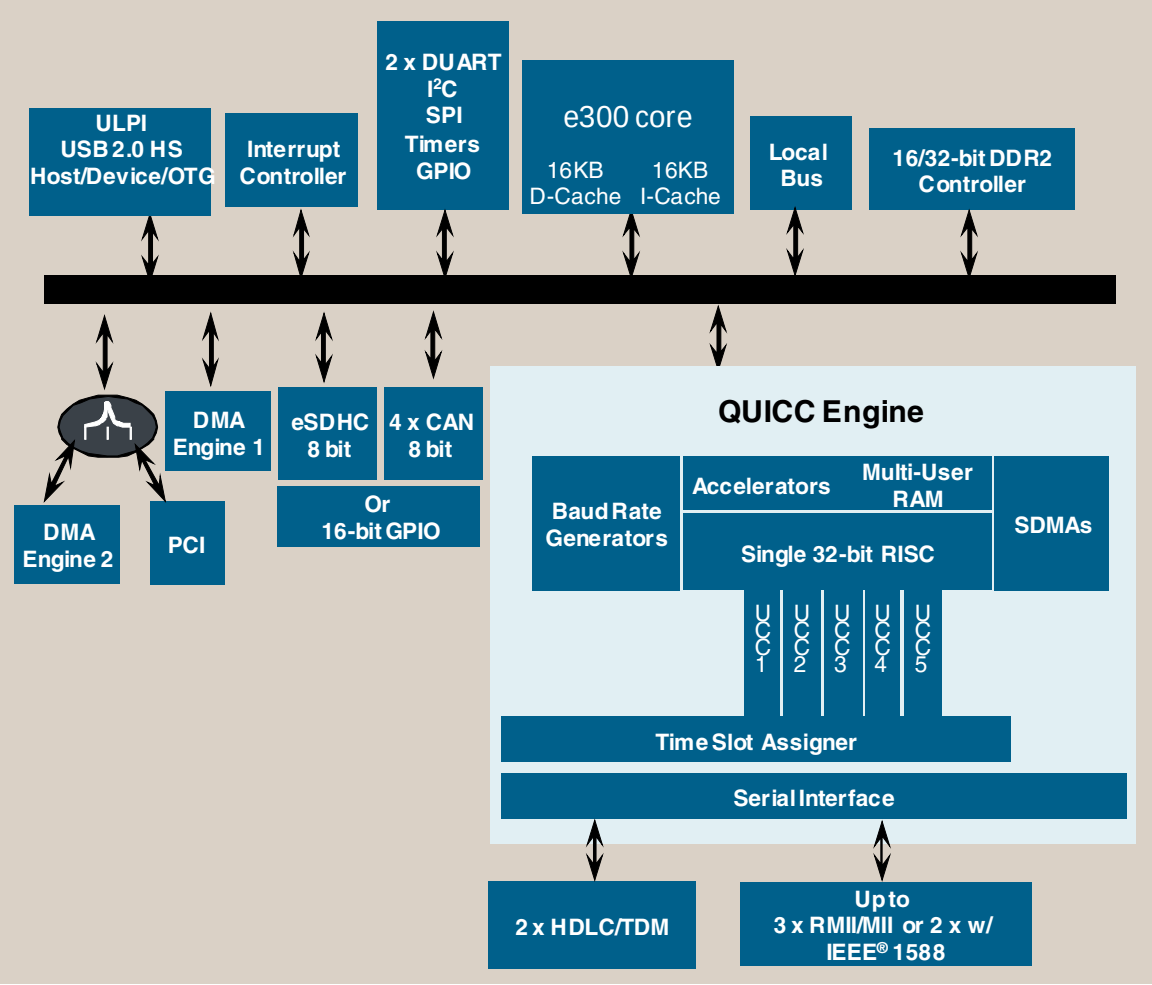

Figure 1. MPC8309 Block Diagram

1.1 e300 Core

The e300 core is a low-power implementation of this microprocessor family of reduced instruction set computing (RISC) microprocessors. The core, based on Power Architecture® technology, implements the 32-bit portion of the instruction set, which defines 32-bit effective addresses, integer data types of 8, 16, and 32 bits, and floating-point data types of 32 and 64 bits.

The core is a superscalar processor that can issue and retire as many as three instructions per clock cycle. Instructions can execute out of program order for increased performance; however, the core ensures that instructions complete in program order.

The e300 core integrates independent execution units including: an integer unit (IU), a floating-point unit (FPU), a branch processing unit (BPU), a load/store unit (LSU), and a system register unit (SRU). The e300c3 integrates an additional integer unit for a total of two IUs. The ability to execute instructions in parallel and the use of simple instructions with rapid execution times yield high efficiency and throughput for the e300 core-based systems. Most integer instructions execute in 1 clock cycle. Having two IUs combined with a reduced multiply latency of 2 cycles means that the e300c3 core is able to sustain single-cycle multiply-accumulate throughput, which is a significant improvement on previous processor versions. In the e300c3 core, the FPU is pipelined so a single-precision, multiply-add instruction can be issued and completed every clock cycle. The e300c3 core provides hardware support for all single- and double-precision floating-point operations for most value representations and all rounding modes.

Figure 2 shows the e300c3 block diagram.

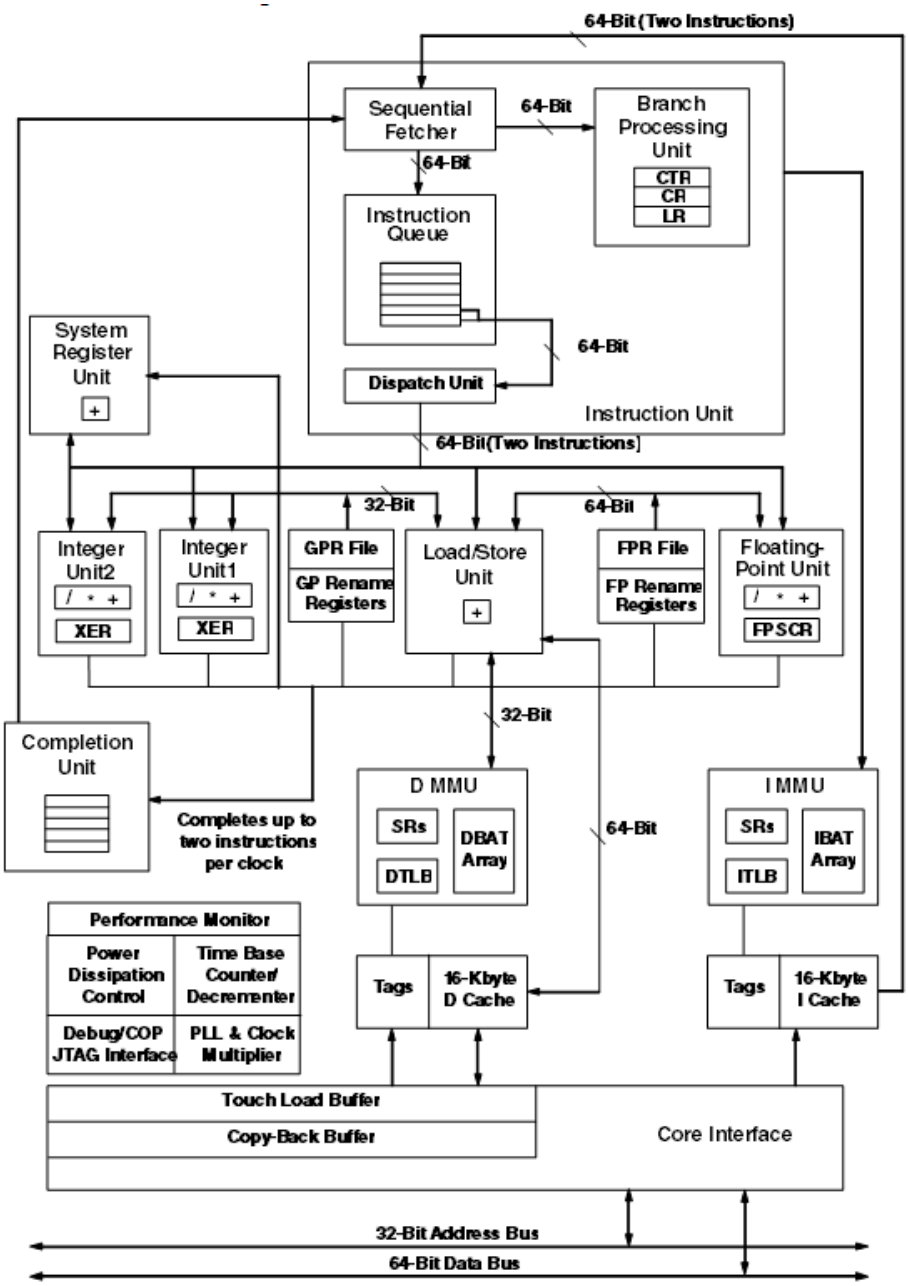

Figure 2. e300c3 Core Block Diagram

The e300c3 includes 16-Kbyte, four-way, set-associative instruction and data caches. The MMUs contain 64-entry, two-way, set-associative, data and instruction translation lookaside buffers (DTLB and ITLB) that provide support for demand-paged, virtual-memory, address translation, and variable-sized block translation. The TLBs use a least-recently-used (LRU) replacement algorithm and the caches use a pseudo least-recently-used algorithm (PLRU).

1.2 QUICC Engine

The MPC830*x* product family offers users a fully programmable, integrated communications processor traditionally not found on entry-level devices. The QUICC Engine has been historically used by the PowerQUICC family to address a wide range of communication and networking protocols with the flexibility to address new emerging standards. The MPC830*x* product family integrates the next generation of the engine found on PowerQUICC II devices, offering improvements while maintaining a high level of compatibility. The unified communications controller (UCC) is the workhorse of the QUICC Engine and currently supports the following protocols:

- 10/100 Mbps Ethernet/IEEE Std 802.3® through MII and RMII interfaces
- IEEE Std 1588™
- HDLC/Transparent (bit rate up to QUICC Engine operating frequency/8)
- HDLC bus (bit rate of up to 10 Mbps)
- Asynchronous HDLC (bit rate of up to 2 Mbps)
- TDM interface supporting up to 128 channels of HDLC and Transparent protocol at 64 Kbps
- UART
- PROFIBUS
- BISYNC (bit rate of up to 2 Mbps)
- Time slot assigner and serial interface (SI) for two TDMs and full duplex routing RAM of 512 entries.
- Ethernet management functionality by Ethernet MDC/MDIO and SPI

Figure 3 shows the internal architecture and the interfaces provided by the QUICC Engine module. A common multi-user RAM is used to store parameters for the RISC engine. The RISC has an Instruction RAM associated with it, which is loaded with the microcode image. The instruction RAM is used to optionally run additional code.

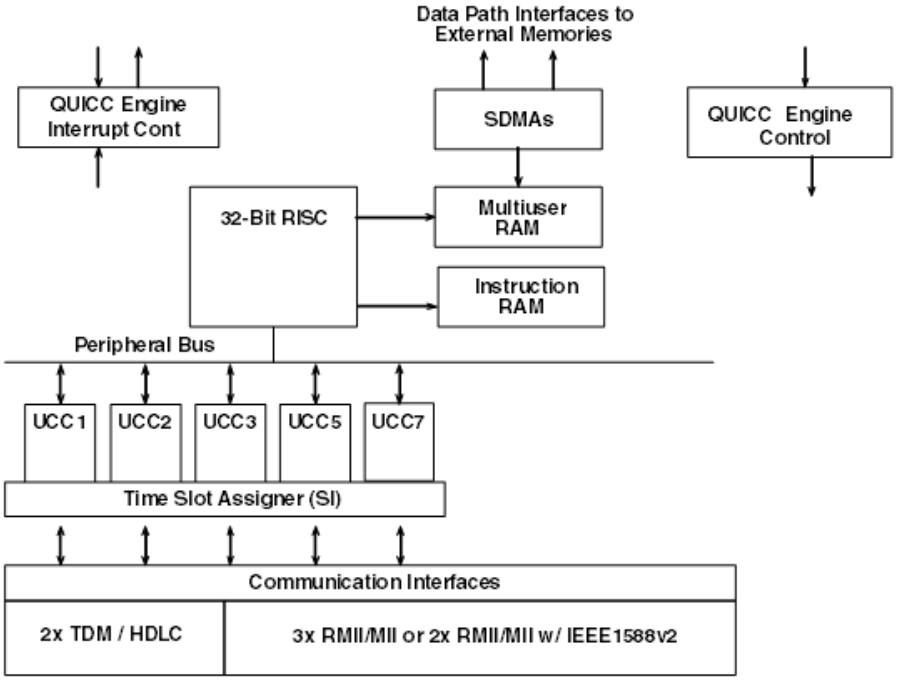

Figure 3. QUICC Engine Block Diagram

1.3 Industrial Interconnect

The MPC830*x* supports interfaces that can be useful in both industrial and factory automation applications. These devices include a DDR2 SDRAM memory controller, two DUARTs, and GPIOs. The MPC8306 adds IEEE 1588v2 for clock synchronization and four FlexCAN interfaces with 64 message buffers each. In addition to IEEE 1588v2 and FlexCAN, the MPC8309 adds 8-bit ECC to the DDR2 memory controller and a 32-bit PCI-2.3 controller.

1.3.1 DDR Controller

The fully programmable DDR SDRAM controller supports most JEDEC standard x8 and x16 (and x32 with 8-bit ECC for MPC8309 only) DDR2 memory available at the 266-MHz data rate. Dynamic power management modes that simplify memory system design and a large set of special features support rapid system debug.

1.3.2 FlexCAN Module

The FlexCAN module is a communication controller implementing the CAN protocol according to the CAN 2.0B specification. It includes up to 64 flexible message buffers, an Rx FIFO for ID filtering that can match incoming IDs, has short latency time, and is backward compatible with previous module version.

1.3.3 DUART Controllers

The included DUARTs support full-duplex operation and are compatible with the PC16450 and PC16550 programming models, and both the transmitter and receiver support 16-byte FIFOs. Each DUART provides a standard, two-wire data (TXD, RXD) for each port that can also be configured as one 4-wire interface (RxD, TxD, RTS, CTS).

1.3.4 IEEE 1588

Hardware-assisted implementation of IEEE 1588 can enable heterogeneous systems that include clocks of various inherent precision, resolution, and stability to synchronize. This protocol supports system-wide synchronization accuracy in the sub-microsecond range with minimal network and local clock computing resources. The hardware assist includes a timestamp unit to recognize PTP frames and transfer relevant timestamps and a real-time clock with high resolution to sub-microsecond accuracy.

1.3.5 PCI Interface

Designed to comply with the *PCI Local Bus Specification*, Revision 2.3, the 32-bit PCI controller is 3.3-V compatible, supports host and agent modes, PCI-to-memory and memory-to-PCI streaming, memory pre-fetching of PCI read accesses, and support for delayed transactions.

1.3.6 GPIO Interface

For general-purpose I/O (GPIO), the MPC8306 supports up to 56 parallel I/O pins, and the MPC8309 supports up to 64 parallel I/O pins that include both dedicated pins and pins multiplexed on various device interfaces. All of the pins support interrupt capability.

1.4 MPC830*x* Features

Table 1 lists the MPC830*x* features by device.

Table 1. MPC830*x* Features by Device

MPC830*x* PowerQUICC II Pro	MPC8308	MPC8309	MPC8306	MPC8306S
Core	e300c3	e300c3	e300c3	e300c3
I-Cache/D-Cache	16 Kbytes/16 Kbytes	16 Kbytes/16 Kbytes	16 Kbytes/16 Kbytes	16 Kbytes/16 Kbytes
Floating point unit	Yes	Yes	Yes	Yes
Core frequency	266/333/400	266/333/400	133/200/266	133/200/266
QUICC Engine subsystem	No	32-bit eRISC	32-bit eRISC	32-bit eRISC
Memory controller	16-/32-bit DDR2 with ECC up to 133-MHz clock	16-/32-bit DDR2 with ECC up to 133-MHz clock	16-bit DDR2 with ECC up to 133-MHz clock	16-bit DDR2 up to 133-MHz clock
Local bus (eLBC)	8-/16-bit up to 66 MHz	8-/16-bit up to 66 MHz	8-/16-bit up to 66 MHz	8-/16-bit up to 66 MHz
PCI interface	No (x1 PCI Express)	32-bit up to 66 MHz	No	No

Table 1. MPC830*x* Features by Device (continued)

MPC830*x* PowerQUICC II Pro	MPC8308	MPC8309	MPC8306	MPC8306S
Ethernet	2x 10/100/1000 MII/RGMII	3x 10/100 MII/RMII or 2x 10/100 with IEEE 1588 v2	3x 10/100 MII/RMII or 2x 10/100 with IEEE 1588 v2	3x 10/100 MII/RMII
USB 2.0	Yes	Yes	Yes	Yes
UART[1]	Yes (2x)	Yes (4x)	Yes (4x)	Yes (4x)
I^2C controller	Dual	Dual	Dual	Dual
SPI	Yes	Yes	Yes	Yes
Interrupt controller	IPIC	IPIC	IPIC	IPIC
IEEE 1588 support	Yes	Yes	Yes	No
eSDHC	Yes	Yes	Yes	No
FlexCAN	Yes	Yes	Yes	No
Package (19 x 19 MAPBGA)	473-pin	489-pin	369-pin	369-pin

1 The UART devices listed in this table are in addition to the UART protocol supported by the QUICC Engine UCCs.

The above devices are built in 90-nm CMOS process technology. They are available in a thermally enhanced mold array process-ball grid array package (MAPBGA). The following highlights the package parameters:

Package outline	19 mm × 19 mm
Package type	MAPBGA
Interconnects (8306/S)	369
Interconnects (8309)	489
Pitch	0.80 mm
Module height (typ)	1.48 mm; min = 1.3 mm and max = 1.61 mm
Solder balls	96 Sn/35 Ag/0.5 Cu (VM package)
Ball diameter (type)	0.40 mm

1.5 Improved Price/Performance with Feature Compatibility

MPC830*x* family provides the ideal migration path for the existing PowerQUICC I and PowerQUICC II families due to its very competitive price/performance ratio and enhanced interfaces to support new protocols in the industrial networking applications. As shown in Figure 4, MPC8306 and MPC8309 are feature-compatible with the legacy PowerQUICC products with improved price-performance characteristics. This allows for significant re-use of software investment while being supported by a strong ecosystem of third party partners and developers.

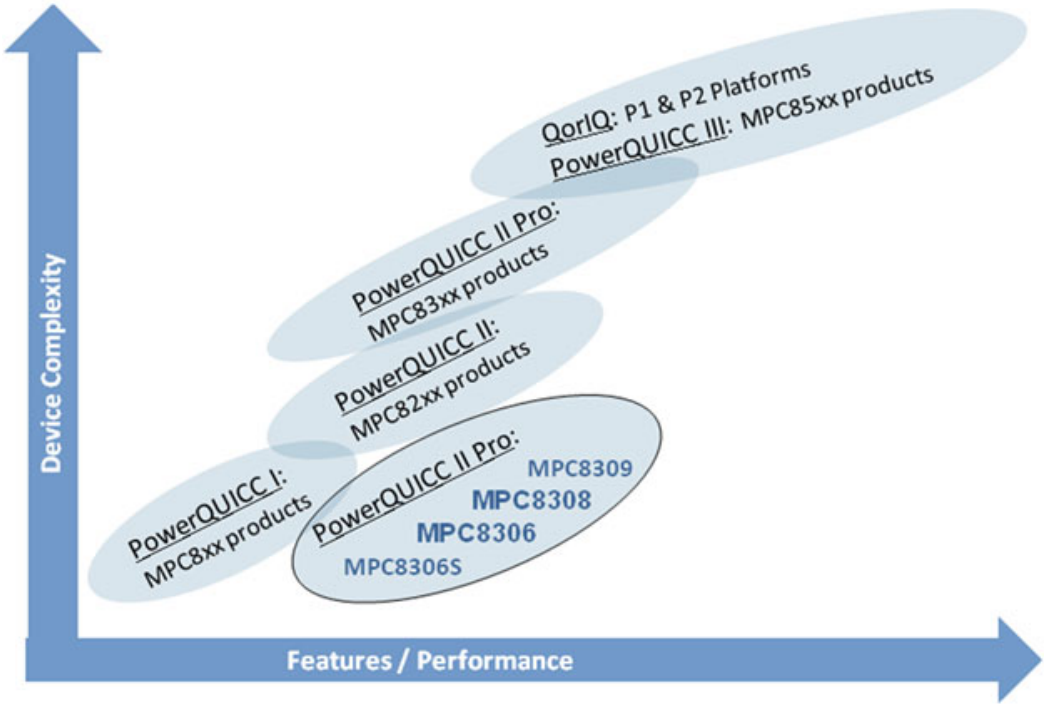

Figure 4. PowerQUICC and QorIQ Price/Performance

The single RISC QUICC Engine on the MPC8306 and MPC8309 supports multiple field bus protocols such as PROFIBUS, CAN/CANopen, and DeviceNET. Leading industrial vendors have also now developed high-level networking protocols to interwork with the field bus protocols across Ethernet. These protocols include EtherCAT, PROFINET, and so on, which can be supported by the MPC830*x* family.

2 Industrial Networking and Control Solutions Based on PowerQUICC

The dominance of Ethernet in the enterprise and telecom networking has created a low-cost, mature cabling and connectivity infrastructure and has increased appeal in other networked worlds. Enhanced mechanisms added by different standards groups have given Ethernet the deterministic qualities needed to operate in these operational and safety-critical environments. Factory networks and building automation systems are adopting Ethernet in complement to and replacement of the various field bus technologies used thus far.

2.1 Single-Chip Solutions for Industrial Applications

Freescale communications processors provide single-chip solutions for industrial applications. In a typical building or factory network there are a number of domains, each with a varying degree of timing and safety constraints. These range from the management and operation levels where standard IT equipment is used on traditional Ethernet and industrial Ethernet-based networks. In the control and field domains, there is a range of nodes and network connectivity options deployed to provide safety, supervisory, process, and logic control over the field equipment. This diverse protocol domain, where both Fieldbus and Ethernet coexist, is the natural preserve of Freescale communication processors. Integration of high-performance

core technology with a diverse range of interconnect and low-level protocol processing options provides the platform flexibility needed to support a broad spectrum of industrial processing nodes. Figure 5 shows a typical factory automation network.

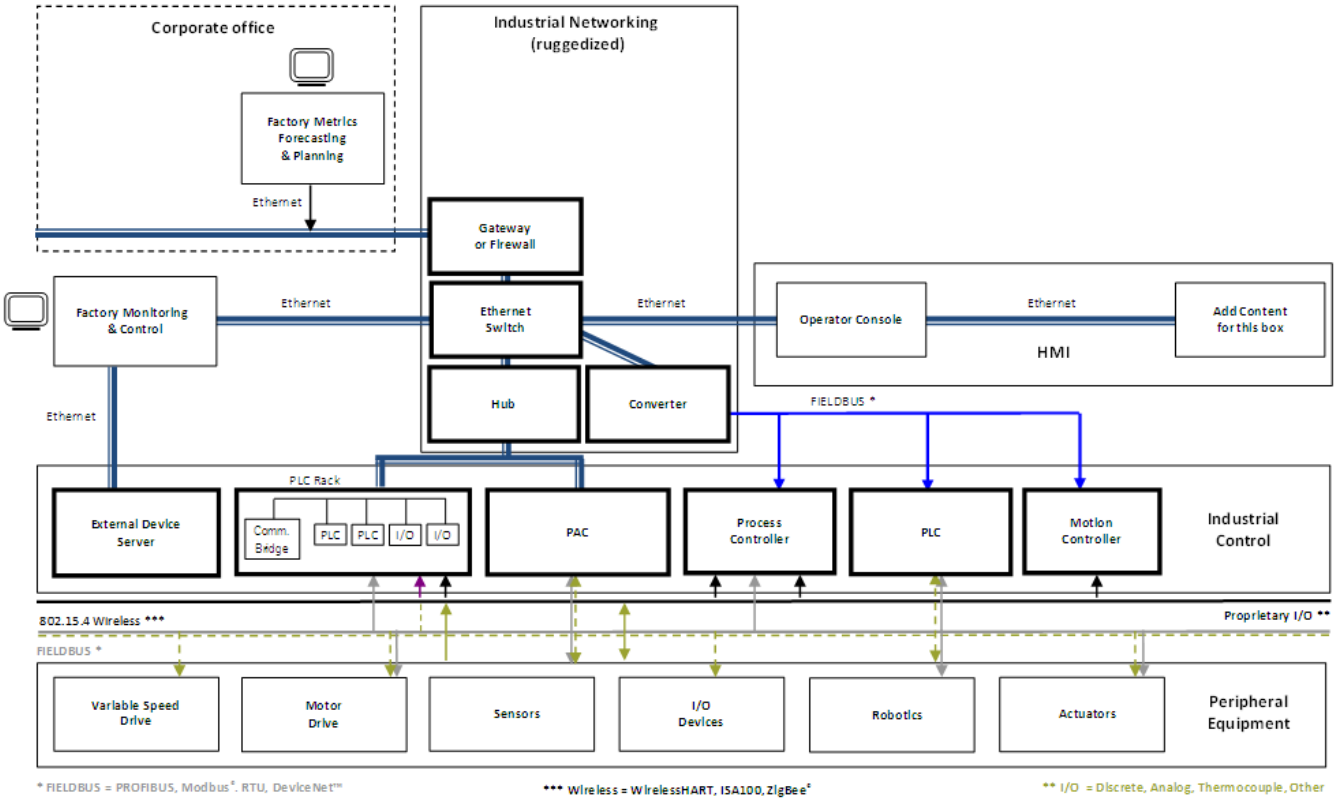

Figure 5. Typical Factory Automation Network

The MPC830*x* can be used in a wide variety of industrial nodes applications, including the following:

- Programmable logic controllers
- Process automation controllers
- Intelligent I/O
- Operator interface terminals
- Drives
- Bar-code and ID systems
- Gateways, bridges and hubs
- And more

The level of programmability and functional integration on MPC830*x* means that the communications protocols and application layer functions can be implemented on a single chip. This has distinct advantages for the end-user, including the following:

- Fieldbus and Ethernet communication on single chip
- Integrated CPU provides flexibility and scalability in terms of performance and scope of application.

- Reduces number of components while boosting available performance
- Fully programmable, which reduces maintenance costs
- A platform for future evolution of applications and communication protocols
- Longevity of supply

3 Use Cases/Applications

Most applications can be implemented using one or a combination of the topologies described in the following subsections.

3.1 Topologies

The MPC830*x* family comprises programmable processors capable of handling enhanced master/slave capabilities. The following diagrams show a master/slave view and an optional connection to upper levels.

3.1.1 Line/Bus Topology

In line or bus topology, each of the signal wires is common to all nodes (slaves and master). A simplified diagram is shown in Figure 6. The protocols supported on the MPC8306 and MPC8309 that can implement this topology are SPI, I^2C, CAN, RS485, TDM, PCI, and Ethernet.

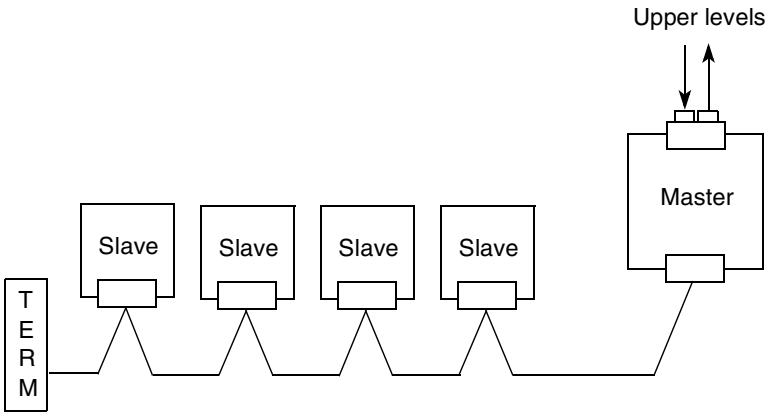

Figure 6. Line/Bus Topology

3.1.2 Star Topology

In star topology, shown in Figure 7, each of the nodes (slaves) has a point-to-point connection to a central node (master). The master can provide routing of data between the slaves as needed. The protocols supported on the MPC8306/309 that can implement this topology, in addition to the protocols detailed in Section 3.1.1, “Line/Bus Topology,” are RS232 and USB, as well as simple GPIOs.

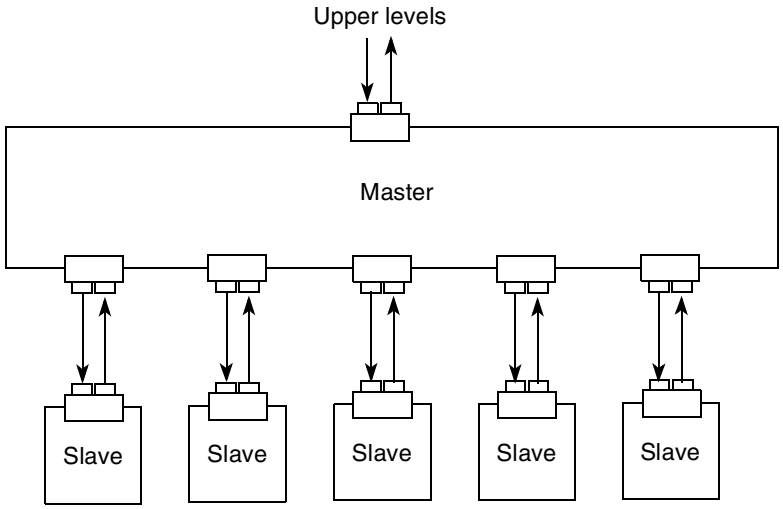

Figure 7. Star Topology

3.1.3 Daisy Chaining Topology

Figure 8 shows master/slave daisy chaining topology. Note that all the nodes in the chain are equal except the last, which must understand that data has reached it but is not addressed to it. This is normally performed by the transport-level software resident on the slave. This is an error condition that must taken care of by the application software. The MPC8306/MPC8309 can be a node in a daisy chaining topology using any of the protocols detailed in the line/bus and star topologies with the correct software.

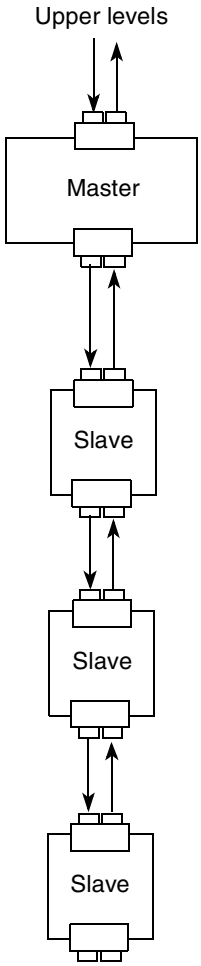

Figure 8. Daisy Chaining Topology

3.1.4 Ring Topology

Ring topology, shown in Figure 9, is an extension of the daisy chaining topology where the end node loops back the data path and completes the ring. With software, the MPC8306/MPC8309 can use any of the protocols detailed in Section 3.1.3, "Daisy Chaining Topology."

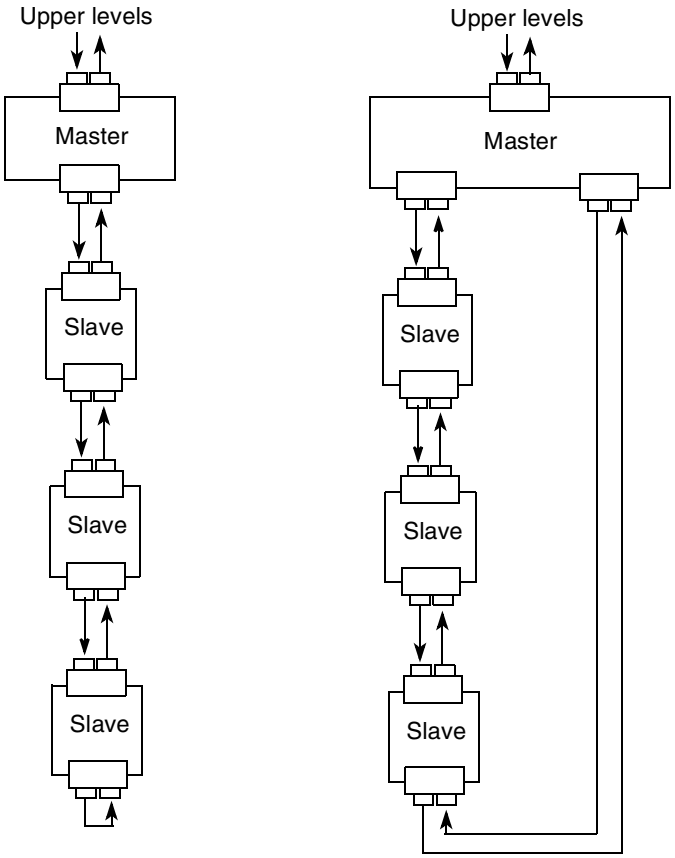

Figure 9. Ring and Redundant Ring Topology

3.2 Protocol Support

A summary of protocols and their interfacing to the MPC8306/MPC8309 and associated topologies is shown in Table 2. Note that the MPC8306/MPC8309 can implement both master and slave nodes in a system.

Table 2. MPC8306/MPC8309 Protocol to Internal Module Mapping

Protocol	External PHY	MPC8306, MPC8309 I/F	MPC8306, MPC8309 Internal Module	Type			Topology			
				Serial	TDM	Ethernet	Line/Bus	Daisy Chain	Star	Ring
CAN	ISO-11898-4	CAN	CAN	√	—	—	√	—	—	—
DeviceNET	ISO-11898-4	CAN	CAN	√	—	—	√	—	—	—
E1/T1	E1/T1	TDM	TSA + UCC	—	√	—	√	—	—	—

Table 2. MPC8306/MPC8309 Protocol to Internal Module Mapping (continued)

Protocol	External PHY	MPC8306, MPC8309 I/F	MPC8306, MPC8309 Internal Module	Type			Topology			
				Serial	TDM	Ethernet	Line/Bus	Daisy Chain	Star	Ring
EtherCAT (Master)	100BaseT	RMII/MII	UCC + IEEE 1588	—	—	√	—	—	—	√
Ethernet/IP	100BaseT	RMII/MII	UCC	—	—	√	—	√	√	—
HDLC	E1/T1	TDM	TSA + UCC	—	√	—	√	—	—	—
HDLC	Level Shifter	HDCL	UCC	√	—	—	√	—	—	—
Modbus TCP	100BaseT	RMII/MII	UCC	—	—	√	—	√	√	—
PROFIBUS	RS485	UART	UCC	√	—	—	√	—	—	—
PROFINET (IO)	100BaseT	RMII/MII	UCC	—	—	√	—	√	√	—
PROFINET (RT)	100BaseT	RMII/MII	UCC	—	—	√	—	√	√	—
PTP	100BaseT	RMII/MII + 1588 I/F signals	UCC + IEEE 1588	—	—	—	—	√	√	—
RS232	Level Shifter	UART	UART	√	—	—	—	—	√	—
RS232	Level Shifter	HDLC	UCC	√	—	—	—	—	√	—
TDM	Level Shifter	TDM	TSA	—	√	—	—	—	—	—
USB	USB	ULPI	USB	√	—	—	—	—	√	—
I^2C	Internal	I^2C	I^2C	√	—	—	√	—	—	—
SPI	Internal	SPI	SPI	√	—	—	√	—	—	—
PCI[1]	Internal	PCI	PCI	—	—	—	√	—	—	—

[1] Available on MPC8309 only

3.3 Protocol Bridging

As the MPC8306 and MPC8309 can implement multiple protocols and perform the master and slave functionally for each, they can also provide a bridging function between similar or dissimilar protocols. For example, a similar protocol bridging would be from PROFIBUS to PROFINET where it is the physical connectivity that is changing. A dissimilar protocol bridging example would be where CAN was converted into a PROFINET. In this case, the conversion would require the complete termination of each protocol to the point where a common mapping of function/telemetry can be achieved (for example, at the abstracted level of smart objects). A matrix of possible bridging combinations is shown in Table 3.

Table 3. Protocol Bridging

	CAN	DevicNET	E1/T1	Ether CAT (Master)	Ethernet /IP	HDLC	I^2C	IEEE 802.1	Modbus/ TCP	PROFIBUS	PROFINET (IO, RT)	SPM	TDM	USB
CAN	—	√	—	√	C	—	—	—	—	—	—	—	—	√
DeviceNET	√	—	—	C	√	—	—	—	—	C	C	—	—	√
E1/T1	—	—	—	—	—	√	—	√	—	—	—	—	√	√
EtherCAT (Master)	√	—	—	—	C	—	—	—	—	C	C	—	—	C
Ethernet/IP	—	√	—	C	—	—	—	—	—	C	C	—	—	√
HDLC	—	—	√	—	—	—	—	√	—	—	—	—	—	√
I^2C	—	—	—	—	—	—	—	√	—	—	—	—	—	√
IEEE 802.1	—	—	C	—	—	√	√	—	—	—	—	√	√	√
Modbus/ TCP	—	C	—	C	C	—	—	—	—	C	C	—	—	√
PROFIBUS	C	—	—	C	C	—	—	—	—	√	√	—	—	√
PROFINET (IO, RT)	—	—	—	C	C	—	—	—	—	√	—	—	—	√
SPM	—	—	—	—	—	—	—	√	—	—	—	—	—	√
TDM	—	—	—	—	—	—	—	√	—	—	—	—	—	—
USB	√	√	√	C	√	√	√	√	√	√	√	√	—	—

Note: √—direct bridge; C—protocol conversion required

3.4 Local Termination

The MPC8306 and MPC8309 can be connected via simple interfaces such as SPI, I^2C, RS232, and even simple GPIO to a series of end-point functions. The rows in Table 4 show interconnect mapping that can be used to make connections for a sample set of endpoints, such as sensors, motors (stepper, AC 1 phase/3 phase, DC), actuators, displays switch, keyboard, and TDM.

Table 4. Local Interconnect Mapping

Protocol	PHY	I/F	Module
SPI		SPI	
I^2C		I^2C	
RS232		UART	UART
RS232		UART	UCC

Table 4. Local Interconnect Mapping (continued)

Protocol	PHY	I/F	Module
CAN	IEEE-11898-4	CAN	CAN
LVTTL		GPIO	
Open Collector			
Open Emitter			
PCI (3.3 V)		PCI	
TDM		TSA	

3.5 Generic Module

Because the MPC8306 and MPC8309 have considerable flexibility, a few modular building blocks can be implemented to cover a large number of applications. An example of the modular building block is shown in Figure 10 and Figure 11 for the MPC8309 and MPC8306, respectively. Note that in configuration 1, TDM is not supported, IEEE 1588 is selected to provide clock and time, and only two Ethernet interfaces are available. Configuration 2, shown in Figure 12 and Figure 13, does support TDM. Table 5 shows the protocols supported for configurations 1 and 2. To obtain all the available combinations, see the applicable device reference manual for configuration information.

Table 5. Module Protocol to I/F Options

Protocol	I/F	Module	Configuration 1						Configuration 2					
			CAN	GPIO	SDHC/ MMC	CAN	GPIO	SDHC/ MMC	CAN	GPIO	SDHC/ MMC	CAN	GPIO	SDHC/ MMC
CAN	Same eight pins	CAN	4	—	—	4	—	—	4	—	—	4	—	—
GPIO		GPIO	—	8	—	—	8	—	—	8	—	—	8	—
SDHC/ MMC		SDHC/MMC	—	—	1	—	—	1	—	—	1	—	—	1
GPIO[1]	GPIO	GPIO	8	8	8	8	8	8	8	8	8	8	8	8
DeviceNET	CAN	CAN	4	—	—	4	—	—	4	—	—	4	—	—
E1/T1	TDM	TSA + UCC	—	—	—	—	—	—	2	2	2	2	2	2
EtherCAT (Master)	UCC	UCC + IEEE 1588	2	2	2	—	—	—	2	2	2	—	—	—
EtherNet/IP	UCC	UCC + IEEE 1588	2	2	2	—	—	—	2	2	2	—	—	—
HDLC	TDM	TSA + UCC	—	—	—	—	—	—	2	2	2	2	2	2
HDLC	HDLC	UCC	2	2	2	2	2	2	—	—	—	—	—	—
Modbus TCP	UCC	UCC + IEEE 1588	2	2	2	—	—	—	2	2	2	—	—	—
PROFIBUS	HDLC	UCC + IEEE 1588	2	2	2	—	—	—	2	2	2	—	—	—

Table 5. Module Protocol to I/F Options (continued)

Protocol	I/F	Module	Configuration 1						Configuration 2					
			CAN	GPIO	SDHC/ MMC	CAN	GPIO	SDHC/ MMC	CAN	GPIO	SDHC/ MMC	CAN	GPIO	SDHC/ MMC
PROFINET (IO)	UCC	UCC + IEEE 1588	2	2	2	—	—	—	2	2	2	—	—	—
PROFINET (RT)	UCC	UCC + IEEE 1588	2	2	2	—	—	—	2	2	2	—	—	—
PTP	UCC	UCC + IEEE 1588	2	2	2	—	—	—	2	2	2	—	—	—
Ethernet	UCC	UCC	2	2	2	3	3	3	2	2	2	3	3	3
RS232	UART	UART	2	2	2	2	2	2	2	2	2	2	2	2
RS232/ RS485	HDLC	UCC	2	2	2	2	2	2	—	—	—	—	—	—
TDM	TDM	TSA	—	—	—	—	—	—	2	2	2	2	2	2
USB	USB	USB	1	1	1	1	1	1	1	1	1	1	1	1
I^2C	I^2C	I^2C	2	2	2	2	2	2	2	2	2	2	2	2
SPI	SPI	SPI	1	1	1	1	1	1	1	1	1	1	1	1
PCI[1]	PCI	PCI	1	1	1	1	1	1	1	1	1	1	1	1

[1] Available on the MPC8309 only

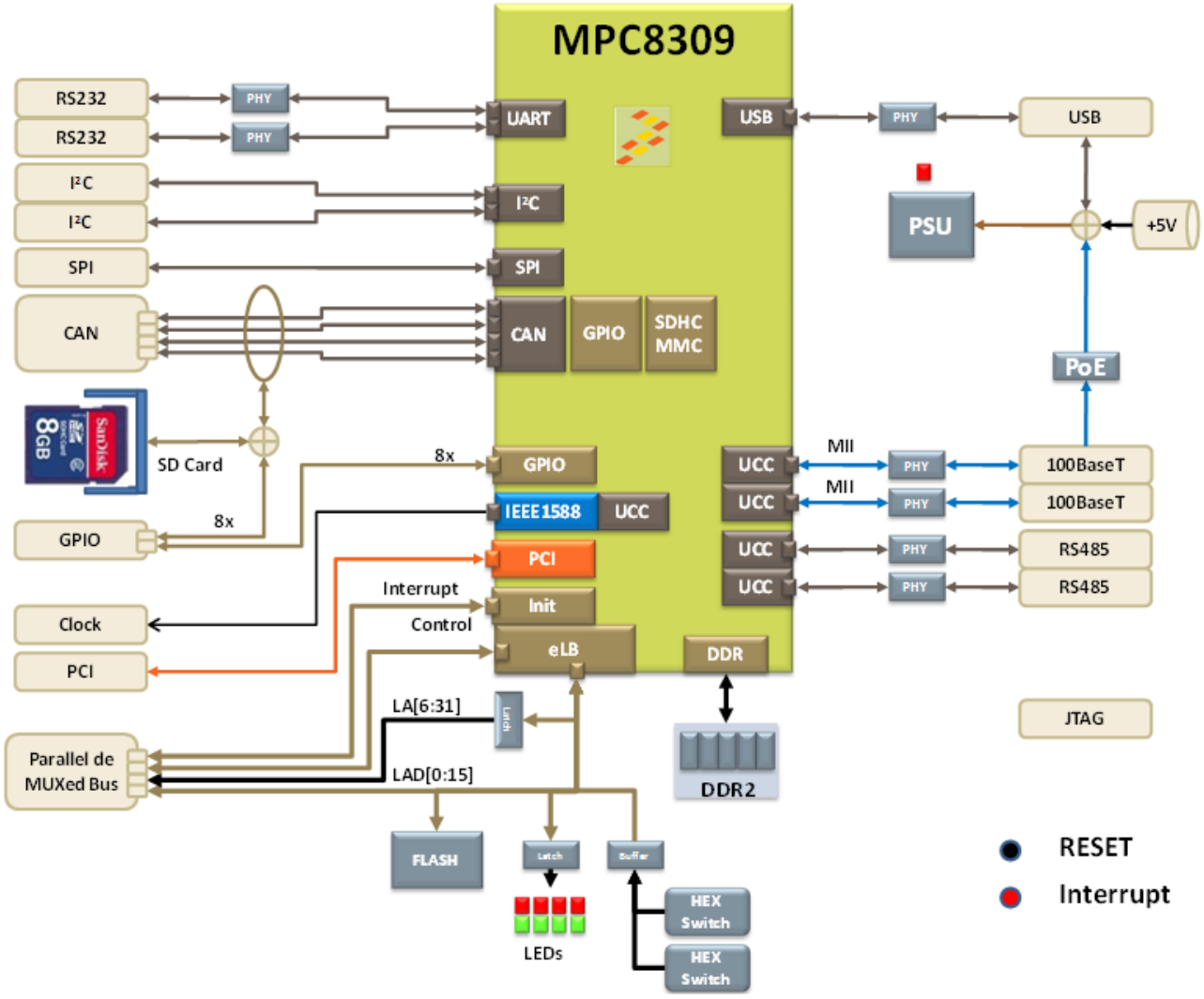

Figure 10. MPC8309 Modular Building Block Configuration 1

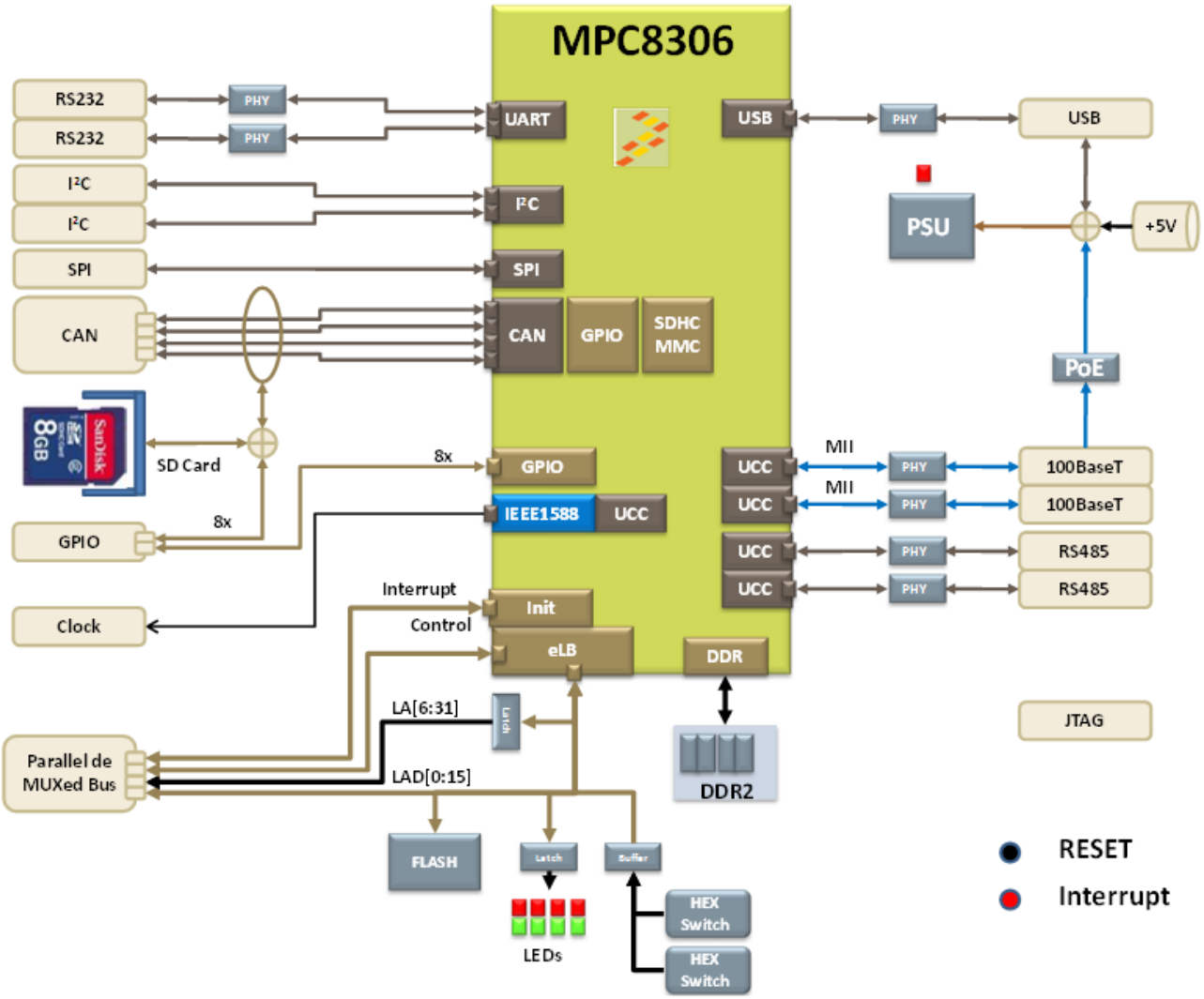

Figure 11. MPC8306 Modular Building Block Configuration 1

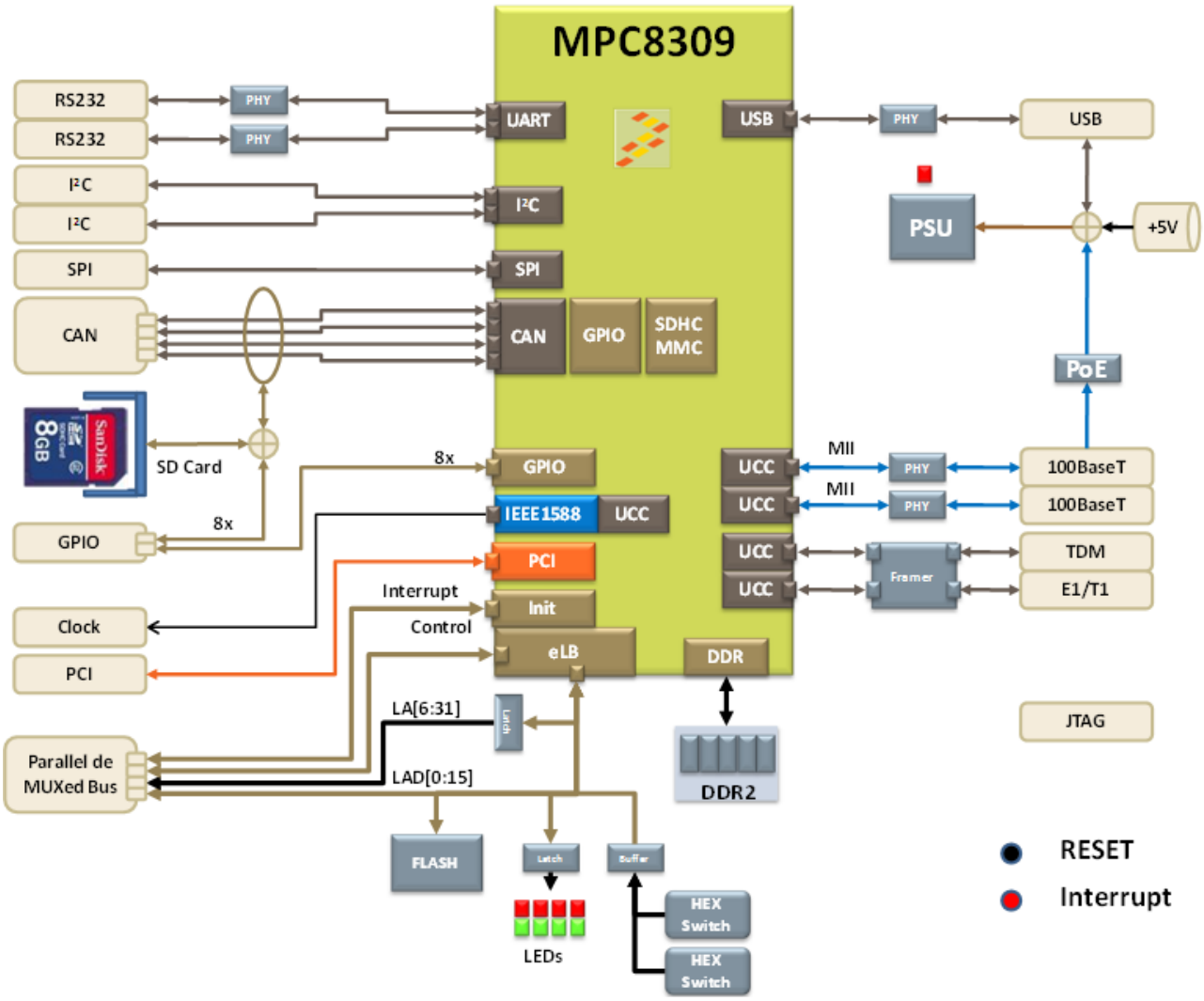

Figure 12. MPC8309 Modular Building Block Configuration 2

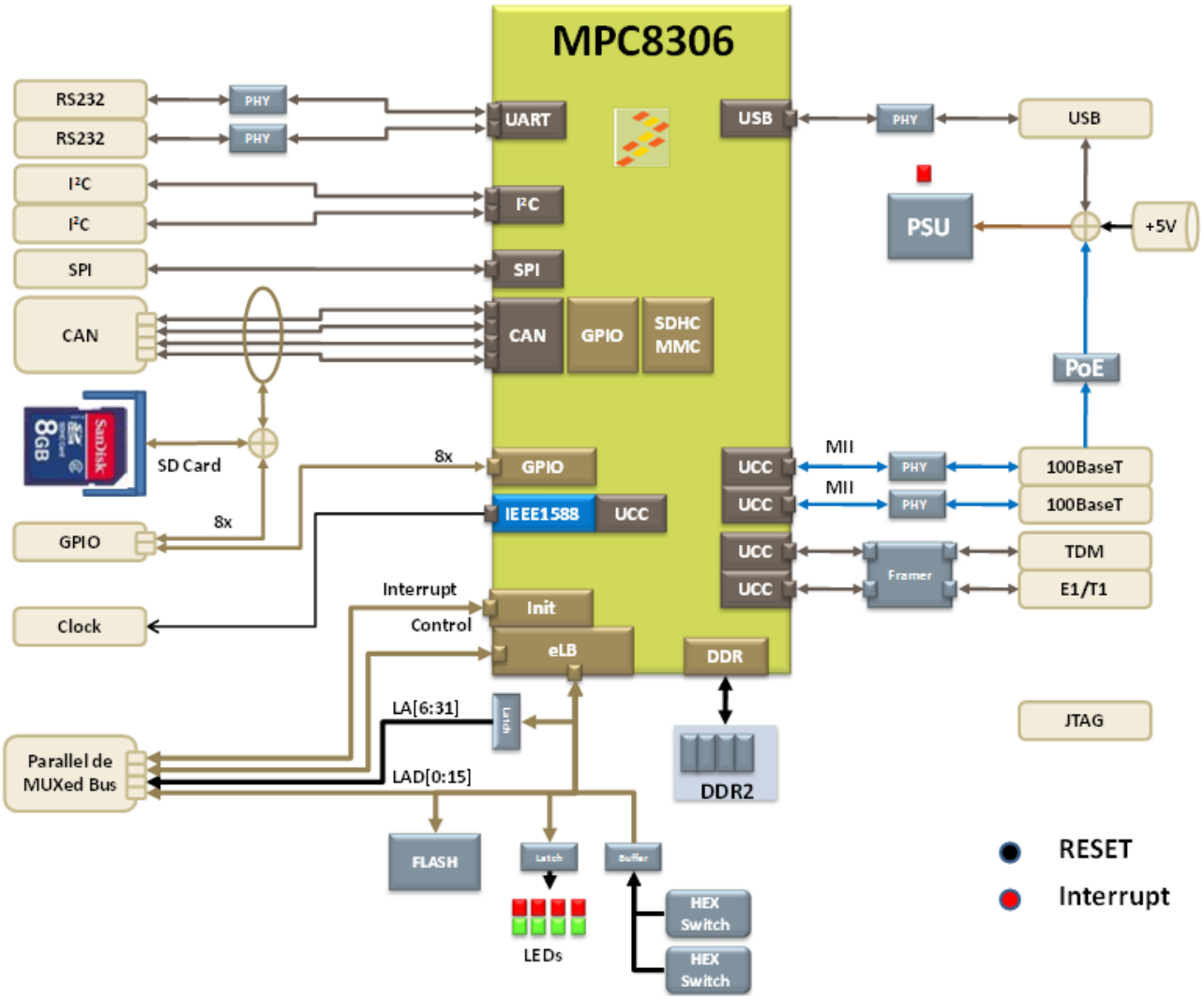

Figure 13. MPC8306 Modular Building Block Configuration 2

3.6 Application Examples

In the following examples using the modules depicted above, the common functions are follows:

USB	Local interconnect for configuration, diagnostics, and telemetry retrieval
RS232/UART	Debug port or simple peripheral connection
HEX switches	Configuration (for example, address, function)

Table 6 shows an example functionality of a slave lift controller, which can be implemented using either the MPC8306 or MPCP8309 module configuration 1. If configuration 2 is used, then a codec can be connected to the framer to provide a phone link allowing voice over Internet protocol (VoIP) to be used for the emergency phone in a human transportation lift.

Further, if the MPC8309 is used, the PCI interface could be linked to a low-cost graphics device providing a human machine interface (HMI). This would allow an operator to locally control the position of each of the lift compartments in a storage system, for example.

Table 6. Example 1—Lift Controller (Slave)

Function	Type	Connection	Protocol	Comment
Master connection	Ethernet	RJ45	PROFINET EtherNet/IO and so on	Connection to central control
Door motor	Motor	RS232	Value	Open/close/speed
Pressure 1	Sensor	I^2C	Value	Door safety switch
Pressure 2	Sensor	I^2C	Value	Height
Temperature 1	Sensor	I^2C	Value	Door motor temperature
Temperature 2	Sensor	I^2C	Value	Lift compartment temperature
Accelerometer	Sensor	I^2C	Value	Movement
Strain gauge	Sensor	SPI	Value	Weight of lift contents
Indicator lights/simple LCD	Output	GPIO	Bit	On/off, character code
Position contacts	Input	GPIO	Bit	Calibration/position reference

Table 7 shows an example functionality of an industrial process control node. In this example, the MPC8306/MPC8309 is acting as the master to multiple PROFIBUS buses (or a redundant implementation), providing all of the local feedback control. It also acts as a slave to a PROFINET link back to the programmable logic controller (PLC) for telemetry and centralized control for safety or report operation.

Table 7. Example 2—Industrial Process Control Node

Function	Type	Connection	Protocol	Comment
Master connection	Ethernet	RJ45	PROFINET	Connection to central control
Slave interface	RS485	RS485	PROFIBUS	Connection to slaves
Local, non-volatile storage	SD card	SHDC/MMC	Memory	Sensor logs, and so on
Indicator lights	Output	GPIO	Bit	On/off

Table 8 shows an example of the possible functionality for remote control hydraulics. This example is completely self-contained and can be implemented by either the MPC8306 or MPC8309 devices. Note that this system could also be a node on a larger system connected via Ethernet or remotely over a phone line (through the E1/T1framer). Additionally, note that the node could also be powered by Power Over Ethernet (POE) if the solenoid's power requirements are acceptable.

Table 8. Example 3—Remote Control Hydraulics

Function	Type	Connection	Protocol	Comment
Solenoid	Actuator	CAN	CAN	Hydraulics
End switch	Sensor	CAN	CAN	End of travel switches (both positions)
Joy stick	Sensor	I^2C	Value	Position control (local)
Key pad	Sensor	I^2C	Value	Configuration (local)
Pressure 1	Sensor	I^2C	Value	Fluid pressure
Pressure 2	Sensor	I^2C	Value	Tip pressure, for example
Indicator lights/LCD	Output	Demuxed eLBC	Bit	On/off codes

Table 9 shows an example of using EtherCAT in master mode to communicate to cascaded slave nodes. Similar to the other examples, a remote link can be provided to this PLC via a telephone link (E1/T1), and if the MPC8309 is used, the PCI interface could be linked to a low-cost graphics device for local display.

Table 9. Example 4—Using EtherCAT to Communicate to Cascade Slave Nodes

Function	Type	Connection	Protocol	Comment
Master to slave connection	Ethernet	2x RJ45	EtherCAT	Either two rings or one redundant ring
Local, non-volatile storage	SD card	SDHC/MMC	Memory	Sensor logs, and so on
Joy stick	Sensor	I^2C	Value	Position control (local)
Key pad	Sensor	I^2C	Value	Configuration (local)
Indicator lights/LCD	Output	Demuxed eLBC	Bit	On/off codes

4 Conclusion

The PowerQUICC II Pro MPC8306 and MPC8309 devices offer multi-protocol termination and interworking functions, integrated with a price-performance optimized CPU core based on Power Architecture technology and a rich peripheral set.

Adopting these devices for industrial control, factory network, and building automation systems allows for a scalable single chip solution that does the following:

1. Is affordable due to the reduced number of components
2. Reduces maintenance costs, because it is fully programmable
3. Protects your hardware and software investment over the long term, because its platform concept embraces future evolution of application and communication protocols.

5 Revision History

Table 10 provides a revision history for this white paper.

Table 10. Document Revision History

Rev. Number	Date	Substantive Change(s)
0	08/2010	Initial public release

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or
+1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064
Japan
0120 191014 or
+81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
1-800 441-2447 or
+1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

Document Number: PQIIPROWP
Rev. 0
08/2010