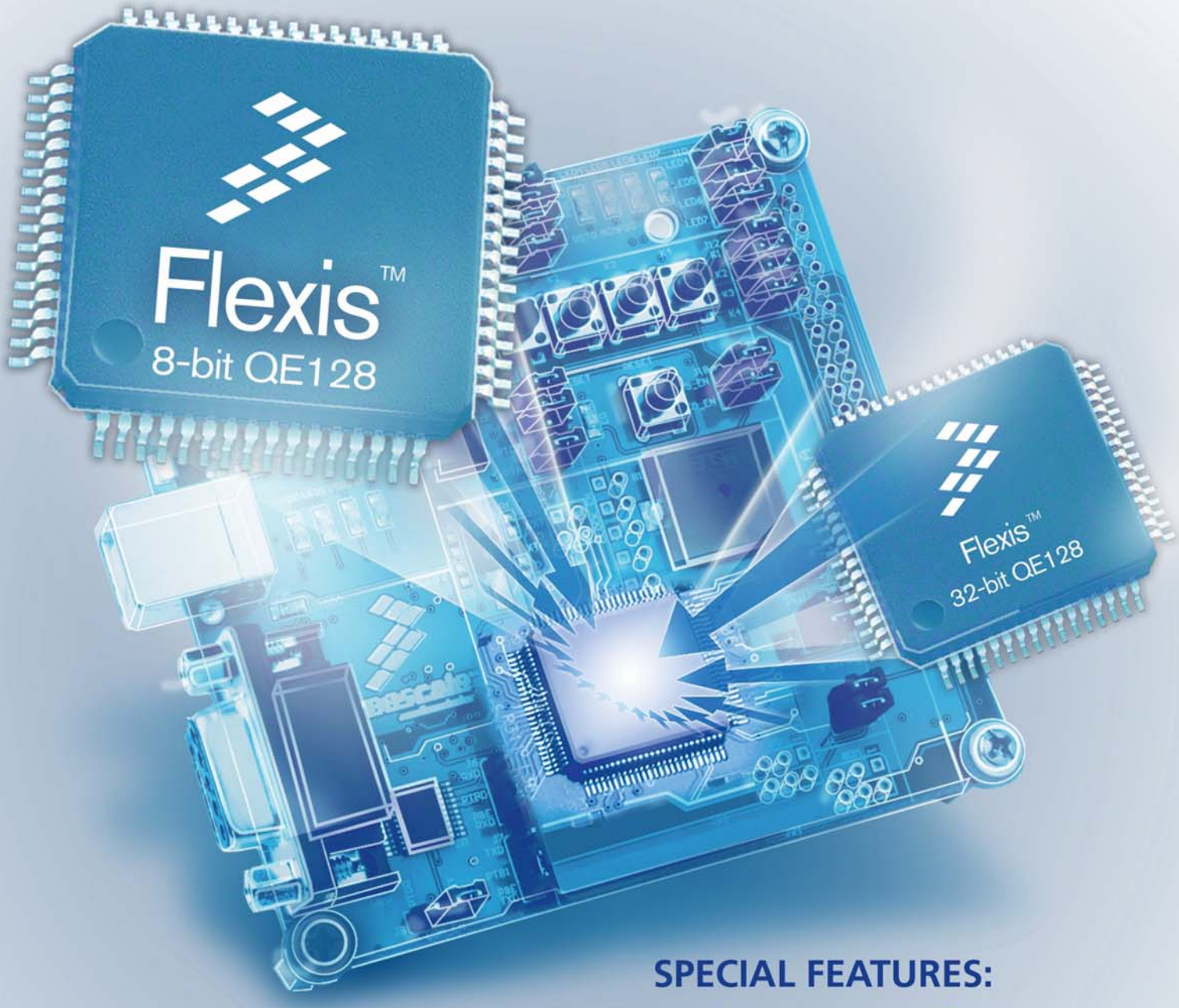
ECE

06/07 July 2007 B53585

Information & Know-how for Embedded Engineers

The Complementary Magazine to

www.Embedded-Control-Europe.com



freescale semiconductor

SPECIAL FEATURES:

- Microcontrollers
- Model-based Design
- Embedded Connectivity
- Analog & Power

COVER STORY:

Flexible migration with complimentary 8-bit and 32-bit microcontrollers

Complete solution for Remote Access Control

Get more at: **www.atmel.com/AVRman**

ATMEL®

Less is More

Geoff Gibson, Managing Director Express Logic UK: "…simpler solutions for less demanding applications."

With much fanfare, and justifiably so, embedded Linux has captured the spotlight in embedded software these days. Linux has earned strong interest and adoption from those in the embedded software development community looking for cost-effective operating system support for their latest embedded device. While Linux offers attractive capabilities, it also is extremely complex, making it relatively difficult to learn and use. But what if low-cost development isn't the goal? What if fast time-to-market demands a much simpler approach? And what if available memory is limited by footprint, cost or power consumption concerns?

In those cases, a multi-megabyte Linux image just doesn't cut it. What's needed is a much smaller, much simpler solution, one that still meets the needs of the project, but doesn't dump additional baggage and complexity on top. This is one of the newer trends in embedded software – simpler solutions for less demanding applications. Consider the development time spent learning which dozen or so of the hundreds of Linux services are needed. Most embedded applications have basic needs, and only use a small subset of the comprehensive capabilities of Linux, maybe just a dozen or two services in all.

Addressing just these basic needs will enable a simple RTOS to satisfy a large number of applications in consumer electronics, medical devices, industrial automation, and wireless networking. Developers of those devices can shortcut development time by choosing an RTOS that addresses their basic needs without excess complexity. And, shortcuts in development time provide dividends in lower development costs, time to market and market share.

The adage "less is more" is supported by findings in a recent survey of embedded developers by Embedded Market Forecasters. The survey reveals that developers using certain RTOSes are more likely to complete their projects on time or even ahead of schedule than users of other operating systems.

Several small, simple RTOS products are available commercially, and most offer full source code, royalty-free licensing, commercial support. These offerings are far simpler and easier to use than Linux. The survey indicates that choosing an RTOS like this, so long as it meets your development needs, is more likely to boost your time-to-market speed over a more complex RTOS selection.

The "less is more" trend will continue, driving other developers away from the Linux solution—not because Linux is not a great technology and ideal for many applications, but because Linux just cannot handle many hard real-time, small footprint, low-cost requirements. These requirements are more and more common, and more suited to one of the small RTOSes on the market.

For fast time to market, truly, less is more.

Cover Photo
Freescale Semiconductor

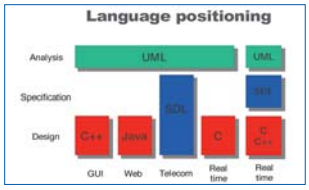
Interfacing a USB flash drive to a PIC microcontroller PAGE 10

This article looks at the detailed implementation and programming of a design to link a low-cost PIC microcontroller to a flash drive via a USB2.0 full speed interface.

Technology trends for highly efficient air conditioners PAGE 16

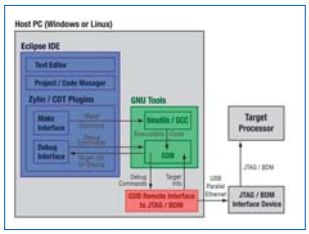
This article describes the developmental trend in power electronics technology for implementing highly efficient air conditioners.

Model-driven design languages for real-time and embedded applications PAGE 18



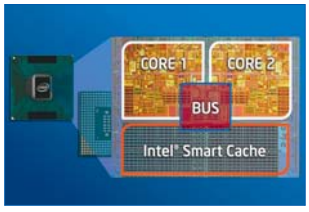
Model-driven engineering is an approach to software development based on abstract models of the system to be developed.

Building an embedded cross-development environment with Eclipse PAGE 24



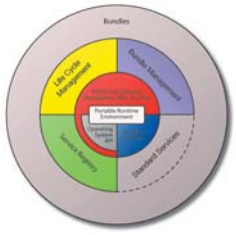
This article describes how to construct a free or low-cost cross-development environment based on the open-source Eclipse IDE and GNU toolsets. The task can be difficult and time-consuming, but the article shows how preconfigured Eclipse projects and pre-built GNU tools help speed construction.

Asymmetric real-time multiprocessing on multi-core CPUs PAGE 31



The latest multi-core processors are ideal for implementing multi-OS embedded applications. Virtualisation technology makes it possible for a multi-core system to easily support multiple operating systems on a single computer platform.

Building applications for network-enabled Linux-based devices PAGE 34



The POCO C++ libraries are a collection of portable, cross-platform C++ class libraries that can be used to create network-centric applications in technical and embedded computing. The core libraries are developed in an open source project.

Make sure they line up for your product first!

PROBLEM SOLVED

Green Hills® • SOFTWARE, INC. •

With more than half of the product development cycle consumed by debugging, finding bugs faster means your product will get to market first.

Green Hills Software provides premier tools that pinpoint the most elusive bugs in minutes, instead of hours or days. With the MULTI® development environment's time-saving code analysis tools, errors in code are automatically found, long before the debugging process begins.

MULTI and the TimeMachine™ debugger allow developers to easily find bugs so that shipping a product with known problems becomes a thing of the past.

With Green Hills Software's sophisticated technology you'll produce a better product and get it out the door long before your competition.

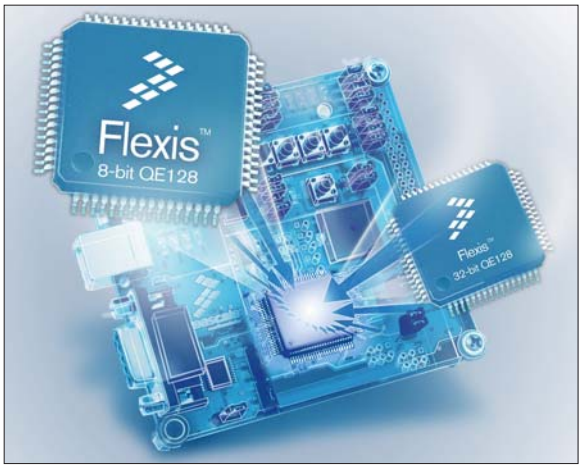
Call or visit us on the web **www.ghs.com** to learn more.

France:	+33 (0)1 43 14 37 00	Germany:	+49 (0)721 98 62 580
Israel:	+972 (0)9 9584060	Netherlands:	+31 (0)33 4613363
Sweden:	+46 (0)46 211 33 70	UK:	+44 (0)1844 267950

Flexible migration with complementary 8-bit and 32-bit microcontrollers

By Inga Harris, Freescale

This article introduces Flexis, which means a single development tool to ease migration between 8-bit and 32-bit MCUs, and a common peripheral set to preserve software investment with pin compatibility wherever practical.



The consumer and industrial markets are demanding greater energy efficiency in new product designs. Portable products need to extend battery life with each new generation, and larger products, such as white goods, require higher energy efficiency ratings to remain competitive. In many cases, replacing mechanical components with electronics improves board power consumption, and additional software control can fine-tune application power performance.

Advances in product components enable engineers to use new concepts and technologies to conquer the power use limitations of earlier products. The microcontroller (MCU) is key to these power reduction innovations for two reasons. Firstly, the MCU itself consumes power, and second, the MCU controls the other system components that also consume power. Therefore, the MCU manufacturers need to incorporate very-low-power features in their latest product designs.

From the RS08 to the highest-performance ColdFirev4 processing devices, the controller continuum provides compatibility for an easy migration path up or down the performance spectrum. The connection point on the controller continuum is where complementary families of S08 and ColdFirev1 microcontrollers share a common set of peripherals and development tools to deliver the ultimate in migration flexibility. Pin-for-pin compatibility between many Flexis microcontroller devices allows controller exchanges without board redesign, and the MC9S08QE128 (S08) and the MCF51QE128 (ColdFire v1) permit exactly that. Flexis means a single development tool to ease migration between 8-bit (S08) and 32-bit (CFV1) and a common peripheral set to preserve software investment between 8-bit and 32-bit with pin compatibility, wherever practical, to maximise hardware reuse when moving between 8-bit and 32-bit.

System designers have to balance the power consumption (the key parameter being Idd measured in amps) and performance (MIPS) to suit application demands. Generally the faster components are clocked, the more power they consume. In an application that is running most of the time, the most efficient solution is one that will run as slow as the system can tolerate. A motor controller is an example of such a system. If the key function is to convert a signal via an analog-to-digital converter (ADC) channel, then the ADC cannot be clocked slower than the minimum sample frequency due to Nyquists law, thus limiting how slow you can run the MCU. In many cases, the application will perform with lowest overall power by performing the scheduled tasks as fast as possible to allow the MCU to quickly revert to a very low-power state. For example, a smoke detector only needs to wake every five seconds, take a reading, make a decision and go back to sleep. This method is very common in battery-operated devices. Such systems are asleep approximately 99 percent of the time. To enable such systems, Freescale has incorporated fast clock switching and fast wake-up routines to help application designers meet their target power consumptions.

Figures 1 and 2 show a smoke detector block diagram and the power vs. time graph. The MCU has one main input from the smoke chamber and two outputs - an alarm and a LED. The real-time clock (RTC) module on the MCU can wake the device periodically from stop 2 mode to enable the op amp to take a reading and process the data to make a decision on the LED and alarm status. The MCU can then re-enter stop 2 mode to conserve power. The HCS08 has a slightly lower stop 2 mode current than the ColdFire V1 microcontroller equivalent part, but the S08 CPU has fewer built-in data processing capabilities. Figure 2 compares the power and time plots of both devices, enabling to choose the solution that best fits an application profile. In applications where task execution speed is of the utmost

Access to hundreds of thousands of quality, name-brand products available online today!*

Quality Electronic Components, Superior Service

Austria	at.digikey.com	Luxembourg	lu.digikey.com
	0800.291.395		800.24.398
Belgium	be.digikey.com	Netherlands	nl.digikey.com
	0800.78.962		0800.020.3710
Denmark	dk.digikey.com	Norway	no.digikey.com
	80.88.13.67		800.15.286
Finland	fi.digikey.com	Portugal	pt.digikey.com
	0800.115.281		800.831.521
France	fr.digikey.com	Spain	es.digikey.com
	0800.161.113		900.983.183
Germany	de.digikey.com	United Kingdom	uk.digikey.com
	0800.1.800.125		0.800.587.0991
Greece	gr.digikey.com		0.800.904.7786
	00800.3122.3456		
Ireland	ie.digikey.com		
	1.800.552.170		
Italy	it.digikey.com		
	800.786.310		

Or visit www.digikey.com for ordering information.

701 Brooks Ave. South • Thief River Falls, MN 56701 • USA

€18.00 flat fee for UPS (United Parcel Service). Express Air for delivery within 1-3 days (dependent on final destination). All Prices are in EURO and British pound sterling and include duties.

*New product added daily.

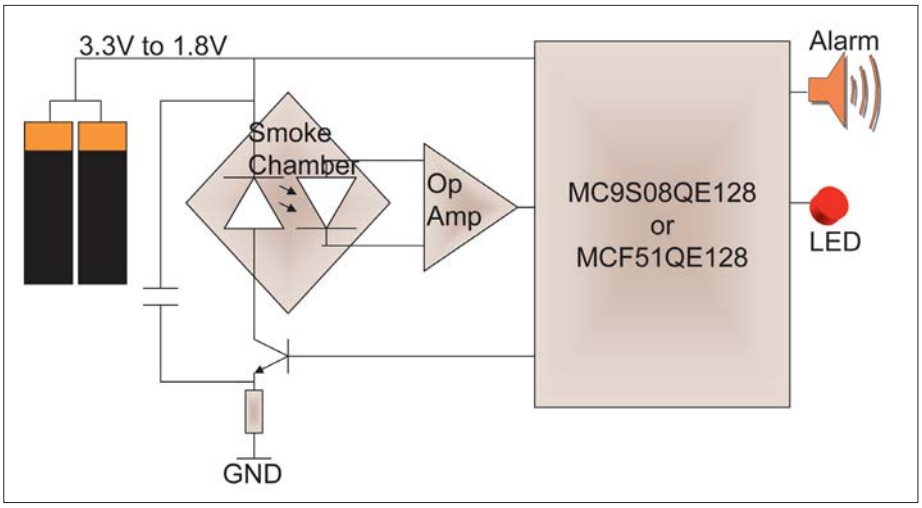


Figure 1. Simple smoke detector block diagram

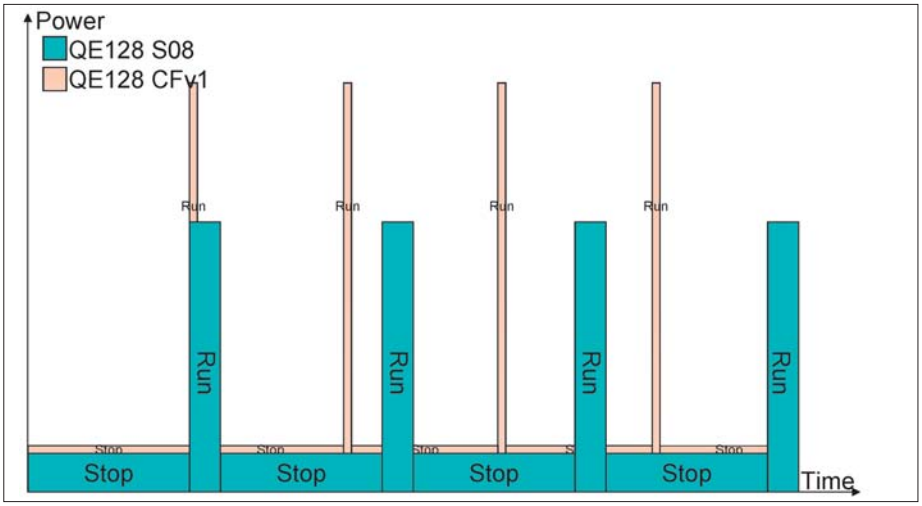


Figure 2. Power vs. time plot for smoke detector application

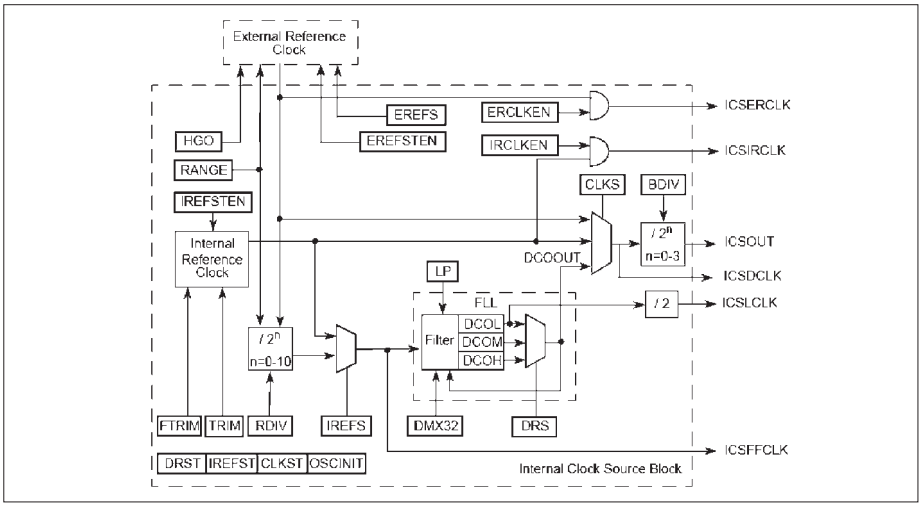


Figure 3. The heart – the internal clock source

importance, the MCF51QE128 ColdFirev1 microcontroller device may be the better solution. The pin compatibility of these two products means that migrating from one strategy to the other is quick and easy, as the same board, software and tools can be used. The heart of the QE128s (both the S08 and ColdFirev1 microcontroller products) is the internal clock source module (ICS) as shown in figure 3. This module enables designers to select an external reference clock (ERCLK) from 32 kHz up to 16 MHz or an internal reference clock (IRCLK) that is trimmable from 31.25 kHz to 39.06 kHz. The heart of the ICS is the frequency lock loop (FLL) block which multiplies its input clock up to a maximum 50 MHz. The input clock to the FLL, also known as the FLL reference clock, must be in the 31.25 kHz to 39.06 kHz range for the FLL to operate correctly. This is simple with the internal FLL reference clock as all you have to do is trim to the correct range. The external reference clock can be as high as 16 MHz and can be divided by a reference divider (RDIV), which is programmable from one to 1024. The FLL can also be bypassed if a low-frequency bus is required. A second divider block, bus frequency divider (BDIV), can divide the clock signal down by one, two, four or eight before it is put out onto the ICSOUT signal. The MCU bus clock is ICSOUT divided by two.

The ICS also controls an independent 1 kHz very-low-power oscillator (LPO), which can be used by the RTC and the watchdog (COP). It is a feature that can be used to conform to EN60730, the standard for automatic electrical controls for household use and similar applications. There are different clock sources which can be used by each module on the MC9S08QE128. Other clocks are OSCOUT, which is a direct path to the external clock and XCLK, which is the signal going into the FLL block - either the internal oscillator or the external clock source, post RDIV. The key to the MCUs flexibility is the ability to use the internal oscillator and the external oscillator source for different modules at the same time. This means by running some modules slower than others, power consumption can be reduced. A secondary power-saving feature related to clocks has been incorporated into the QE128 design. The system clock gating control registers 1 and 2 (SCGC1 and SCGC2) gate on (1) or off (0) the clock source control the clock gating to the timers, ADC, inter-integrated circuits (I2Cs), serial communication interfaces (SCIs), debug module, flash memory, external interrupt request (IRQ), keyboard interrupt module (KBI), analog comparator, RTC and serial peripheral interfaces (SPIs). By gating off the clock to unused modules, precious milliamperes are saved in the MCU's run and wait modes. This feature is especially important on devices like the QE128 because the MCU incorporates many communications modules and timers.

Gating modules on only when needed, and gating them off when the job is done, is a significant power savings technique. The clock is gated on/off immediately after the register is written. However, this method has vulnerabilities that need to be considered:

✓ after a reset all the clocks are gated on, so to keep power consumption down the clocks

	Typical Bus Frequency	Typical Idd (S08)	Typical Idd (ColdFire v1)	Exit Sources	Exit Time
Run	8 MHz	6.5 mA	13.2 mA	n/a	n/a
LP Run	16kHz	22 µA	50 µA	Clear LP bit	n/a
Wait	8MHz	1.8 mA	1.8 mA	any interrupt	instantly
LP Wait	16kHz	3.3µA	3.3 µA	any interrupt	instantly
Stop 3	n/a	450 nA	520 nA	RTC, LVD/LVW, ADC, ACMP, IRQ, SCI, KBI	6ns
Stop 2	n/a	350 nA	350 nA	RTC, IRQ or RESET	29ns

CPU mode comparison chart

should be gated off as soon as possible,
✓ writes to registers associated with a gated off module have no effect,
✓ to avoid erroneous operation, it is good practice to disable the module before gating it off and reinitialize it when the module is gated back on.

As described, the ICS is routed to all the power consuming and component controlling peripherals. This allows designers to choose the power reduction and performance levels they need. All other factors being equal:
✓ using the low-range, low-gain external oscillator is more power economical than using the internal circuitry,
✓ bypassing the FLL, or phase-locked loop (PLL), will save power by reducing the frequency as well as the power saved bypassing the FLL circuitry, and
✓ it is possible on the QE128 MCUs to disable rather than bypass the FLL for further power reduction.

Note that when choosing between the PLL and FLL, the FLL is the lowest power option, although it is less accurate in the long term.

The ICS on the QE128 MCU has six modes of operation: FLL engaged internal (FEI), which is the default mode; FLL engaged external (FEE); FLL bypassed internal (FBI); FLL bypassed external (FBE); FBI low power (FBILP) and FBE low power (FBELP). Each is activated by the LP bit in ICS control register 2. FBILP and FBELP reduce the voltage swing on the oscillator tracks, lowering overall power consumption.

Choosing the right mode for the right stage of the system design can make or break the power consumption targets. The days of simply having the choice of a run mode and a low-power mode are over, as the latest generations of MCU products have a variety of modes with various compromises and advantages. Exit paths, wake-up times and register retention are all choices that you have to make. Along with the usual run mode, the QE128 has five other power saving modes. Three of them, wait, stop 3 and stop 2, are not new to the S08 core and are common in other cores, although they may be under different names.

In wait mode the CPU shuts down purely to conserve power. The MCUs system clocks and full voltage regulation are maintained. Wait mode saves between 30 to 60 percent of the run mode current, depending on the bus frequency, and any interrupt will allow wait mode to be exited instantly. Stop modes halt the system clocks and place the voltage regulator into standby. Stop 3 on the QE128 MCU consumes the highest power of the two stop modes and has the advantage of allowing a wake-up time of just 6ns. There are three versions of stop 3: with BDM enabled, with the voltage regulator active, and with no frills.

Unlike wait, exit from stop 3 is limited to the RTC, low voltage detect/low voltage warning (LVD/LVW), ADC, analog comparator with internal reference (ACMP), IRQ, SCI, KBI and Reset. Stop 2 is the name of the lowest power mode on the QE128 MCU. The internal circuits are powered down while still maintaining the RAM contents and the pin states. The major difference is that there are only three ways to get out of the mode - reset, IRQ and RTC. The MCU resets upon exit in the same manner as a power-on reset.

There are two new power-saving modes on the QE128 device - low power run (LPR) and low power wait (LPW). LPR saves power compared to normal run mode by placing the voltage regulator into standby. To enter this mode, FBELP, the lowest power ICS mode, must be running. The low-voltage protection system must be disabled along with the MCUs internal bandgap and the on-chip in-circuit emulator/ background debug controller (DBG/BDC) modules, as all three modules consume significant amounts of power, regardless of the bus frequency. Since the voltage regulator is in standby, every opportunity to keep the power down in this mode is recommended.

Low power wait (LPW) mode can only be entered from LPR mode and as such the restrictions on LVD circuitry, debugging and ICS mode are the same. The MCU can typically save 50 percent of its current consumption in LPW versus LPR. The table compares the key characteristics of the CPU operation modes.

The QE128 MCU has flash memory which is reprogrammable down to 1.8V, which helps extend battery life in portable applications. Three ports on the QE128 device have set, clear and toggle capabilities, which can speed up pin manipulation, reducing code execution time so low-power modes can be re-entered quickly. All the devices output pins have slew rate control which, when driving external components, can reduce power by slowing transients and drive strength control. ■

Product News

■ Renesas to develop new CPU architecture for MCUs

Renesas announces that it is in the process of developing a new CPU architecture and will offer two CPUs based on this new architecture to address 16- and 32-bit markets, while maintaining compatibility with Renesas existing MCUs. The new architecture will have innovative advances over the M16C and H8S 16-bit CPUs and R32C and H8SX 32-bit CPUs that Renesas currently offers, while offering compatibility with the existing families in terms of CPU instruction sets, peripheral register sets and development tools.

News ID 263

■ Atmel: customizable microcontroller-based SoC platform

Atmel announces a CAP microcontroller-based system-on-chip platform for complex applications that require fast time-to-market. CAP features high-speed local memory, a wide range of industry-standard peripherals and interfaces, and a high-capacity metal programmable block that allows the designer to add a significant amount of custom logic on-chip. CAP is based on Atmel's Metal Programmable Cell Fabric that provides nearly identical routed gate densities to those of standard cells in the same process.

News ID 264

Interfacing a USB flash drive to a PIC microcontroller

By Fred Dart, Future Technology Devices International (FTDI)

This article looks at the detailed implementation and programming of a design to link a low-cost PIC microcontroller to a flash drive via a USB2.0 full speed interface.

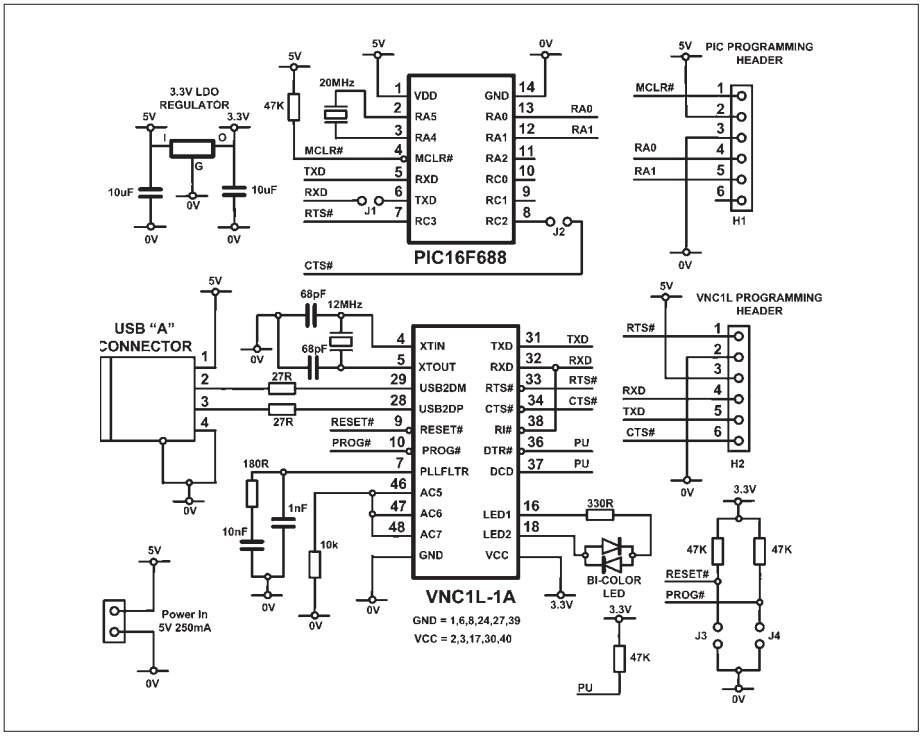
USB interfaces are everywhere today, with low-cost flash memory drives and all kinds of USB peripherals being readily available, but these are very much focused on the PC market. Attempt however to make use of these peripherals in the 8- and 16-bit embedded market, and you find that implementation, cost and power consumption become major considerations. Part of this is down to the embedded controllers that are used in such systems. Devices such as the PIC family of controllers from Microchip are widely used with a broad range of memory densities and peripherals, but they lack the interfaces, resources and performance to incorporate a USB host controller. In this example application, a VNC1L Vinculum controller provides the interface between the PIC as the system controller and a USB2.0 full speed port. This allows a USB flash memory drive connection, for example, to be accomplished with a minimum of implementation time and overhead.

The controller is based around a custom processor core, with twin direct memory access (DMA) engines to accelerate data transfers and a 32-bit numeric co-processor to optimise the calculations for the file system - all in a single chip with 64 kbytes of embedded flash program memory and 4 kbytes of internal data SRAM. Vinculum is specifically targeted at the embedded USB controller market and requires a minimum of external support components. One key feature of the Vinculum core is that its code length is significantly reduced compared with common MCU cores. Reducing the code overhead of the core allows much more functionality to be squeezed into the on-chip e-flash memory. Such features are complementary to a PIC-based embedded system. The schematic of such a system is shown in figure 1, using Vinculum to link a small PIC MCU to a USB "A" connector and hence to a USB flash drive

The PIC is the controller of the system, taking data from sensors or other sources via its general purpose I/O pins (RC0, RC1, RA2 on pins 9,10,11), converting the data format and writing that data in a stream to a file on the flash drive. Commands and data are sent via TXD (pin 6) to the VNC1L RXD (pin 32). VNC1L handles the FAT 12/16/32 file creation and data storage on the USB flash drive communicating with the drive via USB2DM and USB2DP on pins 28 and 29. Data is read from the flash drive via the same pins, and sent from the VNC1L TXD (pin 31) to the RXD (pin 5) of the PIC for use by the system firmware.

The system is controlled by the firmware on the PIC, with the transfers controlled by instructions issued by the PIC and interpreted by the standard firmware on the Vinculum. Whilst that is a simple description of the system, more is required to complete the design. The devices need power, crystals to control their clocks, and they need to be programmed. Using a 20MHz crystal on pins 2 and 3 of the PIC allows for higher baud rates of up to 115200 bit/s in its UART interface as opposed to the maximum 9600 bit/s achievable by using the internal 8MHz oscillator, thus improving the performance of the system.

PIC I/O pins RC2 and RC3 are used by the PIC firmware to simulate RTS/CTS handshake signals with the VNC1Ls UART interface. A 5V regulated PSU at 250mA is required, providing up to 200mA at the USB "A" connector, 25mA to power the VNC1L and 25mA to power the PIC16F688. VNC1L requires a 3.3V supply which is provided by a 3.3V LDO regulator, and has 5V tolerant I/O pins which enable it to connect to the PIC without using level shifters. For low-power applications the VNC1L can be put into a 2mA sleep mode when not required. To



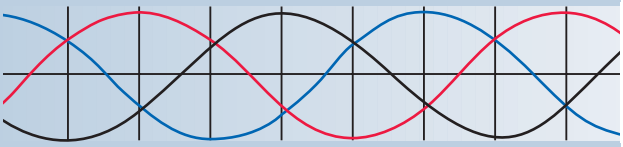
PIC microcontroller and Vinculum USB interface chip

Enjoy the benefits of field-oriented control at lowest system cost

Application kits available now!

INFINEON is the world's first microcontroller supplier to offer optimized 8-bit MCUs capable of supporting Field Oriented Control (FOC), a technique for operating electric motors that results in smooth and efficient operation at all speeds. Implemented on its 8-bit XC886 and XC888 families, Infineon brings FOC motor control solutions to price-sensitive applications such as dishwashers, fan controls and washing machines. This highly innovative solution offers significant reduction of system costs and frees up to 40% CPU capacity, providing plenty of headroom for other applications. Unlike most competitive hard-coded FOC implementations, the XC886/8 microcontroller-based solution offers the added benefit of software reprogrammability to give developers more versatile application options.

Field Oriented Control



Key features of XC886/888 enabling FOC

- 16-bit fully programmable co-processor (vector computer)
- PWM unit with hardware trigger for A/D conversion (CapCom6E)
- A fast 10-bit A/D converter

Key benefits

- 16-bit performance at 8-bit costs
- Higher energy efficiency
- Lower torque ripple
- Excellent dynamic response

Order now at:

www.infineon.com/xc800-foc

infineon

Never stop thinking

wake up the device, strobe the ring indicator pin (RI, pin 38) of the UART interface. If this is connected to the RXD line, as here, it can be triggered by an incoming dummy command to wake up the device. This design also includes a bi-colour status LED indicator powered from pins 16 and 18. This indicates successful enumeration of the USB flash drive and access to the file system. The ViNC1L is programmed with standard firmware, called VDAP (Vinculum Disk and Peripheral) that interprets the commands coming from the PIC. These VDAP commands are DOS-like instructions such as DIR, RD and WR. The command set also supports single byte hex commands which are more suited to control by a microprocessor. VDAP commands are included in the PIC firmware to control access to the USB flash drive. A typical sequence would be to create a file, read/write data to the file and then close the file. This design contains two programming headers, one for each device assuming that a development environment is desired. For a production design, both devices could be pre-programmed prior to insertion on a PCB, thus eliminating the headers and jumpers.

During normal operation, J1 and J2 should be populated and the other jumpers left open. To program the VNC1L, remove J1 and J2 jumpers to isolate the VNC1L UART inputs from the PIC outputs. Disconnect the 5V PSU and then connect a TTL-232R-3V3 cable to H2. Connect the USB side of this cable to a PC with the VPROG programming utility installed. Populate J4 to pull the PROG# pin of the VNCL1A low and temporarily short J3 to reset the device and put it into programming mode. After programming, remember to restore the jumper settings to the operational positions. The programming header for the PIC connects to pins RA0 and RA1 and MCLR# of the device, with the 5V programming voltage/supply being supplied via the header. Disconnect the 5V PSU prior to programming the PIC microcontroller.

The header would be connected to a standard PIC development environment such as a PICKit2, allowing Microchips debug and downloading tools to be used. In the sample C code provided the PIC waits for a flash disk to be detected and then opens a file called hello.txt. The text "Hello World" total with a carriage return and line feed characters are then written to the file. It then closes the file and waits for the disk to be removed. ■

Product News

Microchip: 8-bit microcontrollers with integrated LCD control

Microchip announces six new members of its 8-bit PIC microcontroller family featuring an integrated LCD control module. The PIC18F85J90 family doubles the amount of Flash and RAM memory available in 64- and 80-pin PIC18 LCD devices. This increased memory enables to add more complex functionality, such as speech playback, while taking advantage of the cost savings offered by the PIC18 J series 3-volt manufacturing process.

News ID 296

Gleichmann: 8-bit MCU family with LCD controllers/drivers

NEC Electronics 8-bit 78K0/Lx3 microcontrollers (MCUs) include on chip controllers/drivers for 88 to 288 LCD segments. The MCUs are available with 8 Kbytes to 60 Kbytes of embedded flash memory and RAM from 512 bytes to 2 Kbytes, in pin counts ranging from 48 to 80.

News ID 306

ST: 32-Bit MCU family powered by ARM Cortex-M3 core

STMicroelectronics has announced a family of 32-bit Flash microcontrollers based on the ARM Cortex-M3 core - a core with features specifically designed for embedded applications requiring a combination of high performance (1.25 Dhrystone MIPS/MHz), low power and low cost. The STM32 family benefits from the Cortex-M3s architectural enhancements including the Thumb-2 instruction set to deliver improved performance combined with better code density, and a tightly-coupled Nested Vectored Interrupt Controller for significantly faster response to interrupts, all combined with low power consumption.

News ID 343

Atmel: application processor for Linux-based designs

Atmel announced the AT32AP7001, a member of the AVR32 AP7 family of application processors optimized for cost constrained, Linux-based embedded designs. The device is packaged in a 30 x 30mm VQFP for easy integration into a four-layer PCB design. The processor is built around the AVR32 AP7 application processor core and offers 210 Dhrystone v2.1 MIPS when clocked at 150 MHz and powered at 1.8V (1.4 DMIPS/MHz), with an active power consumption of 75mA (500 µA/MHz).

News ID 320

Renesas: extended M16C microcontroller platform

Renesas announced 12 different M16C/64 group models and 20 different M16C/65 group models for car audio, home audio, and similar consumer products as well as industrial applications, as additions to the M16C Platform microcontroller lineup. These models are upgraded functions and performance extensions of the current M16C/62P group and are compatible with current M16C Platform products, offering flexibility in implementing system functional enhancements and version upgrades.

News ID 352

ST: new version of ARM9-based flash microcontrollers

STMicroelectronics has announced the release of a new version of its 32-bit ARM9-based STR910F flash microcontrollers The enhanced memory accelerator in the STR910FA boosts performance by allowing the burst Flash memory to stream instructions more freely to its ARM966E CPU core. Peak performance remains the same, at 96 MIPS during the execution of sequential instructions, but the average performance has increased significantly when instructions are non-sequential, as a result of the larger memory accelerator.

News ID 368

MIPS: synthesizable 32-bit processor surpasses 1 GHz

MIPS announces the MIPS32 74K cores, fully synthesizable 32-bit processors that achieve operating frequencies greater than 1 GHz in TSMC 65nm process technology. Broadcom has already taken the first license of this innovative design with advanced microarchitecture, enhanced DSP capabilities.

News ID 269

More information about each news story is available on www.embedded-control-europe.com/ece_magazine You just have to type in the "News ID". —

Power supply sequencing with supervisor functions in PMUs

By Todd Vanyo, Texas Instruments

This article discusses the building blocks needed in a multi-channel power management unit (PMU) to implement the supervisor functions, and provides an example using the TPS65023 to sequence the DaVinci processor family.

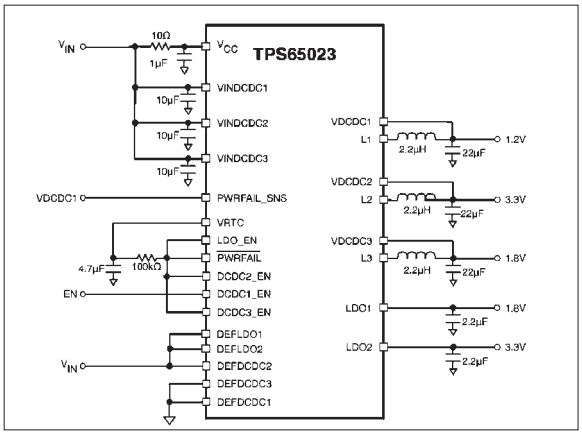


Figure 1. Sequencing schematic of the TPS65023 pins

Most processors, from simple 8-bit microcontrollers to the latest multicore microprocessors, require some form of power supply sequencing. Power supply supervisors have long existed to help solve this problem. But with cost and space at a premium in the ever-increasing number of consumer products containing processors, supervisor ICs may not be welcome. Basic supervisor functions can be implemented with a PMU if the proper building blocks are available. Supervisor ICs are not complicated devices. The basic unit of information is the output of a comparator, which compares an input voltage to a fixed reference voltage – usually generated internally to the supervisor. When the comparator output is low, the supervisor reset output is held in some state (typically low). A timer is started when the input voltage rises above the internal reference and the supervisor reset output changes state when the timer finishes. The timer could be based on the fixed count of an oscillator or the voltage on a capacitor – the latter making the timing variable. The output of the supervisor is usually tied to the reset of the processor or the enable of another power rail.

Supervisor ICs come in a number of different combinations – single voltage sense, multiple voltage sense, fixed reset time, variable reset time, etc. All these provide options for the system designer when designing the sequencing. But incorporating sequencing costs money and board space - it is another part to purchase and place on the board. This leads to the question of how this function can be implemented in a more cost-effective way. The power supply is the best place to implement sequencing. If the PMU design targets a specific processor, internal logic may already exist to sequence each rail as required. In this case, a single enable input to the PMU may trigger a sequence of events that leads to each rail becoming available at the right time. Even a general purpose PMU that is processor agnostic can implement the necessary sequencing, as long as the same supervisor building blocks are available. The TPS6502x family of PMUs has these elementary building blocks available.

There are as many sequencing requirements as there are processors. Some require that the memory and I/O voltage be available before the core voltage and some are the exact opposite, requiring that the core voltage be available first. One positive aspect is that sequencing requirements have become more forgiving and do not require the amount of time they once did. As long as the building blocks are available in the PMU, the necessary sequencing should be possible. By way of example, let's look at the DaVinci family of processors. The only hard sequencing requirement is the need for the ARM microcontroller core (1.2V domain) to come up before the memory and I/O domains (1.8V & 3.3V). The memory and I/O voltages may be come up in any order, even together, after the ARM core is powered. The DaVinci reset signal, /RESET, must be held low until all three voltage domains are fully up. To eliminate the supervisor for this processor line, the PMU must have at least one comparator with an internal reference voltage to monitor the core voltage. It must also have a means of signaling that all three rails are fully up.

The TPS65023 has two independent and standalone comparators – one with the input LOWBAT_SNS and output /LOWBAT, one with input PWRFAIL_SNS and output /PWRFAIL. The internally generated reference is 1V ±1%. The outputs are the open-drain inversion of the comparator output. One comparator is all that is needed for the DaVinci processors, as the memory and I/O can be ganged together. Using two comparators allows you to sequence three rails separately. In addition to the comparator, you need some way to determine that the rails have reached the proper voltage. PMUs accomplish this with an output that indicates that all enabled outputs are within some regulation range. The TPS65023 consists of three step-down converters and two linear regulators that default to the voltages needed by the DaVinci family of processors. Each converter has a Power Good comparator enabled

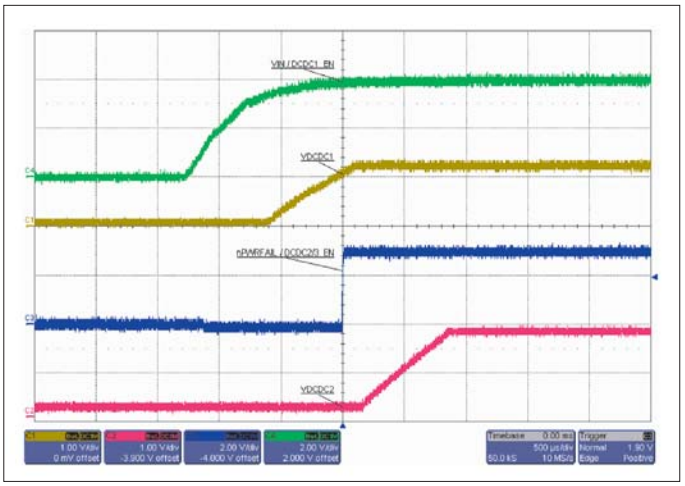


Figure 2. Sequencing trace of the power up of DCDC1 and DCDC2

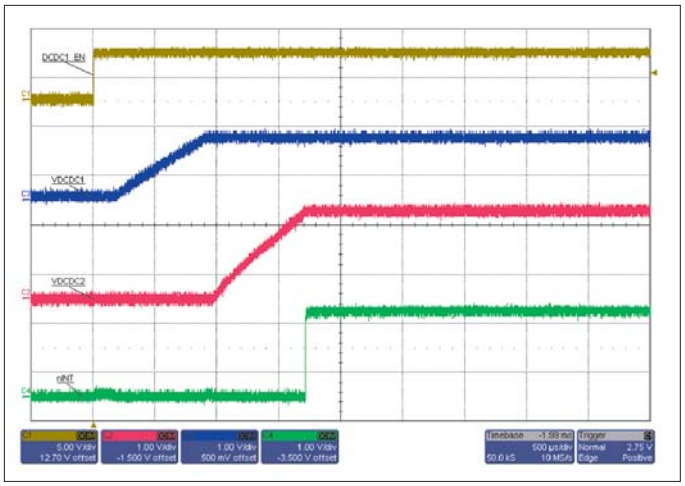


Figure 3. Scope trace of /INT signal for DaVinci Reset

when the converter is enabled and toggles when the output voltage is 95% of the target voltage. The /INT pin is an open drain output that is the logical AND of the Power Good comparator outputs for each enabled converter. When a converter is enabled and its output voltage is more than 10% below the target voltage, the /INT pin will be pulled to ground. To fulfil the DaVinci sequencing requirements, the 1.2V output from the TPS65023 must be up before the 1.8V and 3.3V outputs. One way to achieve this would be to connect the DCDC1 output directly to the DCDC2_EN, DCDC3_EN, and LDO_EN inputs. However, the digital VIH of these input pins is 1.2V minimum, which means there will be insufficient headroom in the event of transients on the DCDC1 output. This is a common issue and the VIH voltages should be carefully checked. Instead of tying the output of the DCDC directly to the other enables, one of the comparators described can be used to compare the 1.2V output to the internal 1V reference voltage. With the output of the comparator pulled up to a voltage that is always present (such as the voltage needed to power an RTC) and tied to the other converter enables, the sequencing requirements would be met. Figure 1 gives a schematic view of the connections to get the necessary sequencing (note that not all TPS65023 pins are shown, only those needed for sequencing the converters). Figure 2 shows the scope trace of the power up of DCDC1 and DCDC2 using the connections in figure 1.

The DaVinci reset requirement, that the /RESET input be held low until all the voltages are available, can be met by using the Power Good logic that drives the /INT pin on the TPS65023.

Since /INT will stay low when an enabled converter is out of regulation, the /INT pin can be pulled up to the 1.8V converter (/RESET on the DaVinci family is a 1.8V input) and tied directly to /RESET. This works because the /INT pin will be low while DCDC1 ramps up. When DCDC1 hits approximately 1V (still below valid regulation), DCDC2 and DCDC3 are enabled – thus holding /INT low until all converters are within regulation. Figure 3 shows the scope trace with the /INT signal (labeled nINT). ■

Ripple management on hysteretic DC/DC controllers

By Ernest Bron, National Semiconductor

This article discusses ripple. Why it is required with Constant On-Time controllers? What impact has it on the output voltage rail? How can a system be designed that has ripple, but with a minimum of negative side effects?

■ Over the last few years numerous hysteretic type DC/DC controllers have been introduced to the market place. Hysteretic control requires no loop compensation which makes them relatively easy to design with. A major disadvantage however is the fairly unpredictable operating frequency combined with the inherent need for an output voltage ripple. The introduction of Constant On-Time (COT) controllers largely eliminated the frequency issue. However basic control in COT-type controllers is still voltage-based, hence a ripple is still required. The basic principle of COT-type controllers revolves around sensing the output voltage ripple and switching the FETs appropriately. A typical controller requires about 10mV of ripple on the feedback pin. To show the effect of this requirement on the output voltage regulation, assume we have a step-down regulator with a feedback voltage of 1.2V and an output voltage of 5V. Here 10mV ripple on VFB translates to about 42mV of output voltage ripple on the 5V output (about 0.8%). This would not normally be problematic for a typical 5V system, however as output and feedback voltage requirements decline, ripple becomes more of an issue. We do the same calculation for a system with 0.8V feedback voltage and 1.3V output voltage, still assuming a minimum ripple of 10mV on VFB. On the output rail we

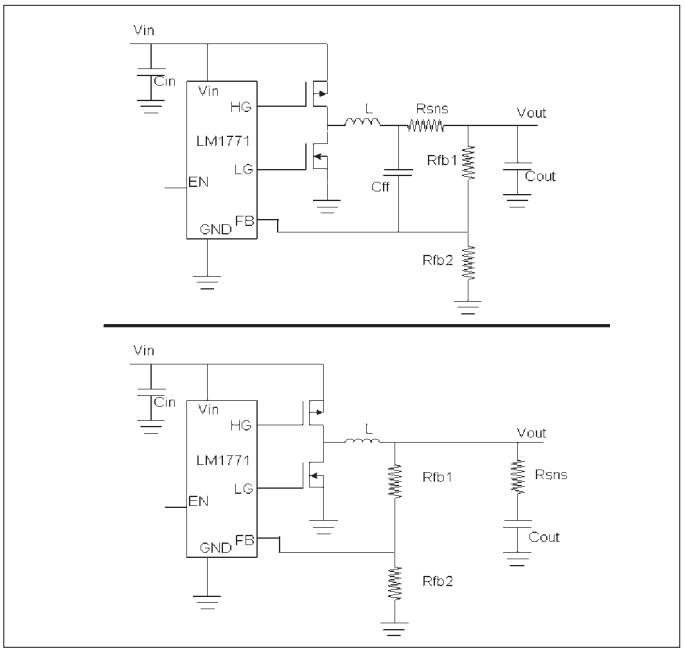


Figure 1. The two basic principles for ripple creation: The top circuit is suited for high output voltage and low output currents. The bottom circuit is suited for low output voltages.

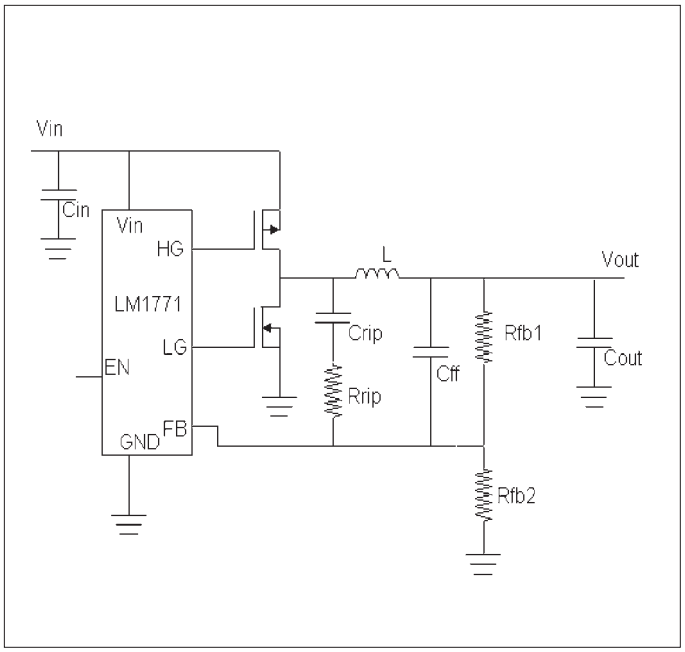


Figure 2. This block diagram shows a circuit that can inject ripple onto the feedback pin with minimal impact on either efficiency or output voltage ripple.

now see a minimum ripple requirement of 16.5mV, or about 1.3%. Combine this with the fact that lower power rails typically require even tighter voltage regulation, and we see ripple does indeed become an issue.

Apart from this, there is also the fact that a design using a COT controller needs to make sure there is a ripple on the feedback pin to start with. Normally ripple is generated by the application load current in conjunction with the output capacitor and its associated equivalent series resistance (ESR). Therefore, a COT controller, or any hysteretic-based controller for that matter, will have a hard time operating properly if the output capacitor has low ESR. Here lies the core of the problem for many designs, because in many cases the use of ceramic capacitors is a must.

Because ceramic capacitors are low cost and offer very low ESR, which helps in reducing voltage spikes in the system, they have become the prevalent capacitor type of choice to use. Most modern power conversion devices such as LDOs or DC/DC controllers have their internal control loop optimised for operation with low ESR output capacitors like ceramic capacitors. Hysteretic-based controllers still require a ripple to function though. To make them work with ceramic capacitors we need to create a ripple. The easy way to create ripple is to 'add' ESR to the output capacitor. Refer to figure 1. Here we see the LM1771, a recently released synchronous COT step-down regulator. It is a good example of all the advantages of using COT-based controllers. The part automatically adjusts the on-time inversely with the input voltage, making the operating frequency constant over input voltage. Because there is no need for loop compensation, there are also no associated pins. The result is a very small-form-factor, easy-to-work-with step-down controller, with very few pins and constant operating frequency.

There are two basic principles for ripple creation which are also discussed in the datasheet. If the output voltage is relatively high (compared to VFB), the top circuit of figure 1 can be used. Here Rsns is placed in series with the inductor to add ripple to the output. Together with Cff ripple it provides ripple to the feedback pin without generating additional voltage ripple or DC offset at the output. In effect Cff provides a low impedance path for the high frequency ripple to go directly to the feedback pin. Care needs to be taken with respect to output currents. As output currents increase the power loss in Rsns is no longer irrelevant. This circuit should be limited to designs with output currents of 2A or less. As the output voltage decreases, the top circuit of figure 1 starts having problems because now the ripple on the output becomes more prevalent. Since the output voltage ripple is out of phase with the ripple directly injected by Cff this causes problems. We now need to move over to the configuration as shown at the bottom of figure 1. Here Rsns is placed directly in series with the low ESR ceramic output capacitor. Ripple in induced by the additional resistance and feed via the feedback circuit (RFB1/RFB1) into the feedback pin. No more need for Cff, but as discussed, we do now see the entire required ripple on the output voltage rail. In effect we are emulating the use of a non-ceramic capacitor with higher ESR. Depending on the application, this may or may not be an issue.

The circuit shown in figure 2 illustrates an alternative to the more commonly provided solutions from figure 1. It offers ripple injection combined with little loss of efficiency and low output voltage ripple. Here ripple is injected from the FET/inductor node. Crip provides a low impedance AC path for current to flow towards the feedback pin. Rrip sets the current level and Cff integrates that current into a voltage ripple placed on top of the DC feedback voltage. There are advantages to this alternative configuration. Firstly there is no additional resistance in the main power path, meaning minimal efficiency loss. Secondly there is no series resistance on the output capacitor, allowing output voltage ripple to remain as low as possible. ■

Technology trends for highly efficient air conditioners

By Guho Jung, DaeWoong Chung, and BumSeok Suh, Fairchild

This article describes the developmental trend in power electronics technology for implementing highly efficient air conditioners.

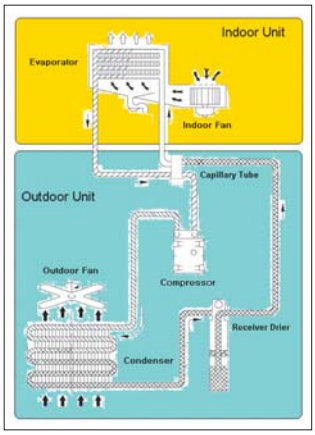


Figure 1. The structure of the general separated type of air conditioner

Home air conditioners are generally divided into two types, the window type and the separated type. In the former all the functions are implemented in one unit, while in the latter the indoor and outdoor units are separated. However, the specifications of air conditioners have recently been upgraded and now cover systems. In the air conditioner system, several indoor units are connected to one outdoor one for installation in large houses and offices. The greatly increased cooling capability has caused summertime power breakdowns due to the excessive use of instantaneous electric power. Consequently tremendous efforts to improve energy efficiency have been made to solve this problem. Additionally, high efficiency is an important factor in the process of replacing CFC refrigerants with natural ones to eliminate the environmental pollution problems caused by CFC refrigerants.

In this situation, each country has tried to tighten its energy regulations. For example, the U.S. Department of Energy announced an enforceable seasonal energy efficiency rating (SEER) standard of 13 for residential central air conditioners which started in January 2006. This represented a 30 percent increase in energy efficiency compared to the previous10 SEER standard. Meanwhile the Japanese government has announced the need for 20% higher efficiency than present levels in order to meet the Kyoto Protocol from 2010, particularly in air conditioners and refrigerators. This means energy saving through high efficiency has become a pivotal consideration in the world of air conditioners, and numerous technologies are currently being developed in response. Moreover new technologies are being developed for reducing noise and vibration generated by outdoor equipment, and network technology is being applied to the air conditioner system. However, the power-saving technologies are the primary concern of this article.

Figure1 shows the structure of a general separated type of air conditioner and the principle of the typical refrigerants circulation cycle. In outdoor units, refrigerants at low pressure are compressed and the resulting heat is radiated away. In indoor units, the ambient heat is absorbed into radiated refrigerants by an evaporator. In this structure the compressor and fan are directly related to the electrical power. The compressor consumes the largest amount of energy of the two. A highly efficient air conditioner is implemented by performance improvement of the heat exchanger and compressor. The performance of the heat exchanger is improved by improving such mechanical technologies as the structure of radiation fin, cooling fan and flow design. On the other hand, motor design and control technologies are needed with designing the compressor, which consumes extremely large amounts of electrical power. The motor for the compressor is located in a compressor with high internal pressure and so the actual installation condition is poor. Single-phase induction motors, robust and needing no control, have been applied for most compressors up to now. But improving efficiency using single-phase induction motors has reached its limit. As an alternative, the BLDC motor (brushless DC) has been applied for highly efficient compressors, its efficiency being higher than that of the single-phase induction motor. But when the inverter efficiency is taken into consideration, the overall efficiency of the BLDC motor and inverter is almost the same as a single-phase induction motor.

The reason for applying the BLDC and its controller to the compressor in spite of its higher material cost is that high efficiency can be obtained by using the variable speed capability over the entire operating range. The variable speed operation is possible by using a three-phase induction motor and inverter. However, the power efficiency in low-speed operation is reduced abruptly in that case and so there is a limit in making a highly efficient compressor using a three-phase induction motor and inverter. Figure 2 shows two different slot shapes of stator used in the distributed and concentrated wiring method. In the first approach to applying a BLDC motor to the compressor, the

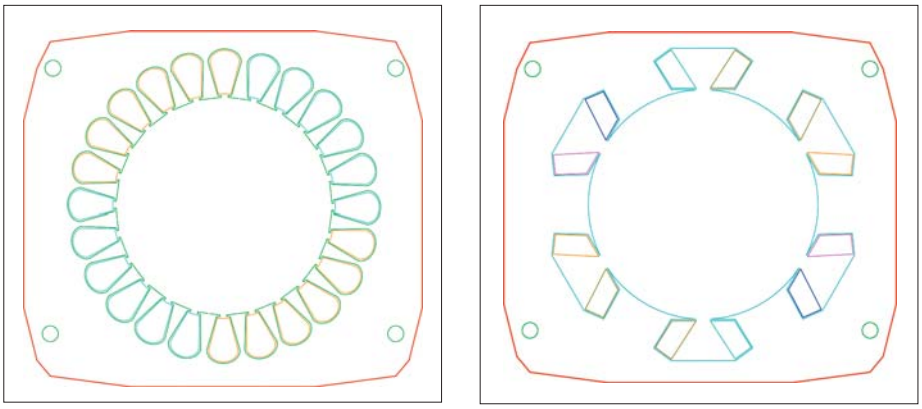


Figure 2. Stator structure of BLDC motor in compressor: (left) distributed wiring method, (right) concentrated wiring method

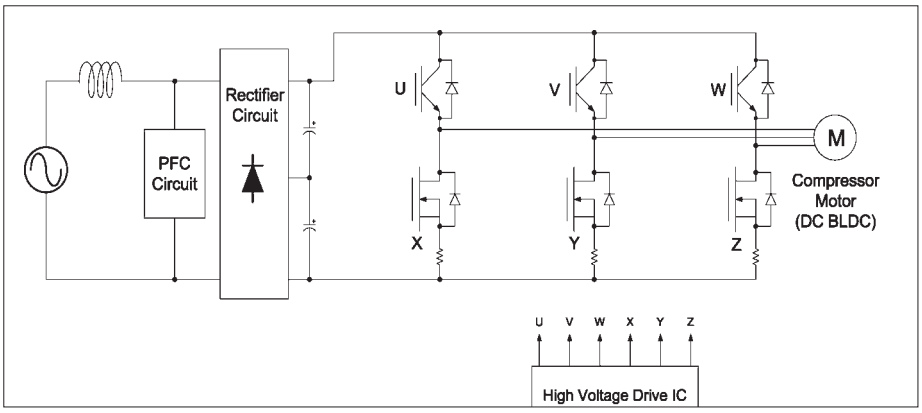


Figure 3. A new inverter using super junction MOSFET

stator wired by distributed wiring method shown in Figure 2a and the two-phase excitation by a rectangular method are mainly adopted. This rectangular method can be implemented by a simpler controller structure compared to the sinusoidal method for driving a PMSM (Permanent Magnet Synchronous Motor), and so was appropriate for home appliance products sensitive to material costs. However, the stator wired by the concentrated wiring method illustrated in figure 2b resulted from strenuous efforts to raise efficiency and cut the product cost by simplifying the manufacturing process. This type of BLDC motor is now increasingly used as standard technology in Japan by air conditioner manufacturers. Even though the concentrated method has some advantages in high efficiency and simplification of the manufacturing process, increased torque ripple has a bad effect on the compressor vibration. To solve this, a partial three-phase or sinusoidal three-phase excitation method is currently being adopted. The best result can be obtained by using conventional vector control, but it is not easy to implement due to the limit imposed by material costs. Accordingly, a partial three-phase excitation method or tuning table method for achieving an output pattern similar to vector control is being tried when large torque ripple arises.. A highly efficient variable speed motor such as a BLDC is used for a highly efficient system in applications with indoor and outdoor cooling fans. However, its contribution to energy savings is smaller because these parts consume relatively less power than the compressor. The energy-saving standard in air conditioners has changed following the JIS (Japan) revision in 2005. Since most air conditioners operate at about half rated capability, energy consumption efficiency at this actually used region is much more important than at the rated one.

Also, inverter efficiency depends on the performance of the power switching device used. So a new highly efficient inverter technology using a super junction MOSFET structure with low on-resistance as the switching device has recently been developed. Figure 3 shows the power circuit of the proposed new highly efficient inverter using the super junction MOSFET. The circuit uses an IGBT for the upper-side element and super junction MOSFET for the lower-side element. By applying a 2-phase modulation method, the low-side current conduction rate can be much higher than the high-side, resulting in a sufficient improvement in efficiency.

Power factor correction circuits are needed for power efficiency improvement, reliability improvement, and compliance with IEC international harmonics regulations. There are two types of PFC, active and passive. In this article, two active methods, partial switching and full switching PFC are explained because they have been popular in several air conditioner applications. In the waveforms of partial switching PFC, the first objective for deciding the IGBT on- time is to pass the harmonic regulation of AC input current. A first idea is to use a look-up table for the IGBT on-time according to the output power. The output power is calculated as below by the product of three-phase voltage and current vectors.

Pout = Vas*Ias + Vbs*Ibs + Vcs*Ics.

When the look-up table is used, the DC link voltage is around 280V under a full range of load conditions. The circuit for full switching PFC is the same as that of partial switching PFC. However the input current has sinusoidal waveform due to pulse width modulation (PWM) at high frequency. ■

Product News

■ Linear Technology: 14-bit 2.8Msps ADC for automotive

Linear Technology introduces the LTC1403AH, a 14-bit, 2.8Msps SAR ADC guaranteed over the automotive -40 C to +125 C temperature range. The LTC1403AH operates from a single 2.7V to 3.6V supply, draws only 14mW at 2.8Msps, and is available in a tiny 10-lead MSOP package.

News ID 255

■ LTC: USB power manager with three integrated buck regulators

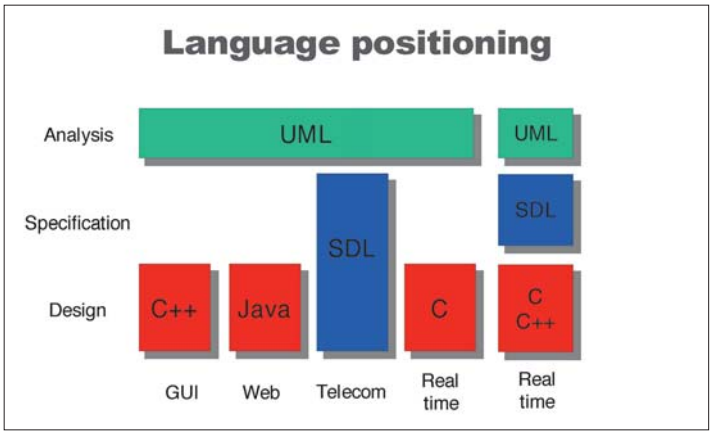
Linear Technology announces the LTC3555, a multi-function power management solutions for Li-Ion/Polymer battery applications. The LTC3555 integrates a switching PowerPath manager, a standalone battery charger, an ideal diode, I2C control, three high efficiency synchronous buck regulators plus an always-on LDO, all in a compact, low-profile 4mm x 5mm QFN package. The LTC3555's PowerPath control feature seamlessly manages power flow between an AC wall adapter or USB port, Li-Ion battery and system load while its 'instant-ON' operation ensures system load power even with a dead or missing battery.

News ID 240

Model-driven design languages for real-time and embedded applications

By Emmanuel Gaudin, PragmaDev

Model-driven engineering is an approach to software development based on abstract models of the system to be developed.



The best-known MDE approach is the model-driven architecture defined and registered by the object management group (OMG) in 2001. Another well-known approach is model-driven design (MDD) where the focus is narrowed in order to work on a more precise model. A model-driven approach distinguishes three types of models: an abstract model of the system under development called the platform-independent model (PIM), a model defining the platform called the platform definition model (PDM), and an implementable model of the system called the platform-specific model (PSM). The PIM is the system under development and the PDM defines the rules to transform the PIM into a PSM. So in practice the team works on the PIM, the PDM is defined by the application domain or the company, and the PSM is automatically generated out of the PIM and the PDM.

To be efficient, the PIM must be abstract enough to be independent of the platform on which the system will be implemented, but at the same time it should be precise enough to be translated into a PSM. So in order to successfully translate the model, the PIM relies on a virtual machine the characteristics of which are a number of basic services and a semantic strong enough to be expressive. In the 80s, the International Telecommunication Union standardised a language to describe telecommunications protocols: the specification and description language (SDL), under reference Z.100. The main goal was to describe the protocols in an unambiguous way so that all manufacturers' implementations of a standard protocol would be compatible with each other (e.g. a Nokia GSM phone would work with an Ericsson base station). The European Telecommunications Standardisation Institute has extensively used SDL to describe telecommunication standards and it is obvious to state that compatibility has been successfully achieved. The SDL standard is regularly updated and major new versions have been produced every four years since 1984. The 1988 (SDL '88) version was considered to be the first usable version, the 1992 version introduced object orientation, and the 2000 version aligned SDL with some UML concepts such as the class diagram.

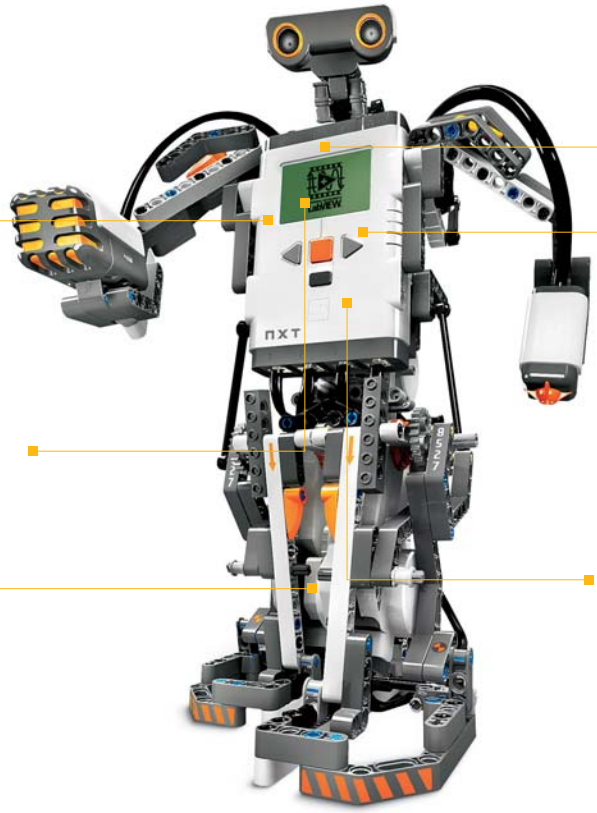
The main characteristics of SDL are: it is a graphical language, it is object oriented, it is event oriented, the model is independent from implementation, the language defines a strong semantic of execution and it contains abstract data types. Because it embeds abstract data types and a syntax to manipulate data, SDL models are formal (complete and non-ambiguous). An SDL model can be fully described because of this characteristic but it does not have to be. It is possible to describe non-deterministic models with, for example, the use of the ANY keyword allowing to describe "any" input or "any" execution path. It is also possible to leave out some operations with the use of undefined "OPERATORS" describing an operation interface and leaving the implementation to the designers. An SDL model can also contain informal operations written in natural language. So depending on the level of precision within the model, an SDL system can be informal or very precise. SDL has built-in concepts and services such as processes, messages, timers, and procedures. All these concepts are supported by most RTOS making implementation on a real target straightforward.

The strong semantic of SDL and its built-in services describe an SDL virtual machine on which a model is based. This is actually the main characteristic of the platform-independent model. The definition of eventual external operators, and the implementation of the SDL services provided by the SDL virtual machine constitute the actual definition of the platform, the platform definition model. From the PIM and the PDM, it is possible to fully generate the platform-specific model in an executable language such as C code. As SDL became increasingly used in telecommunications systems, it appeared its built-in concepts were pretty close to the ones used in real-time operating systems:

- the system is decomposed into tasks running concurrently,

NI LabVIEW.

Limited Only by Your Imagination.



Build and program robots with LEGO® MINDSTORMS® NXT using software powered by NI LabVIEW

Communicate via multiple protocols including Bluetooth

Graphically program concurrent, real-time applications

Develop your human machine interface (HMI) display

Independently control multiple servo motors

Target 32-bit microprocessors and FPGAs

Real-Time and Embedded | Signal Processing | High-Performance Test | Industrial Control

PRODUCT PLATFORM

LabVIEW Real-Time Module

LabVIEW FPGA Module

LabVIEW Embedded Development Module

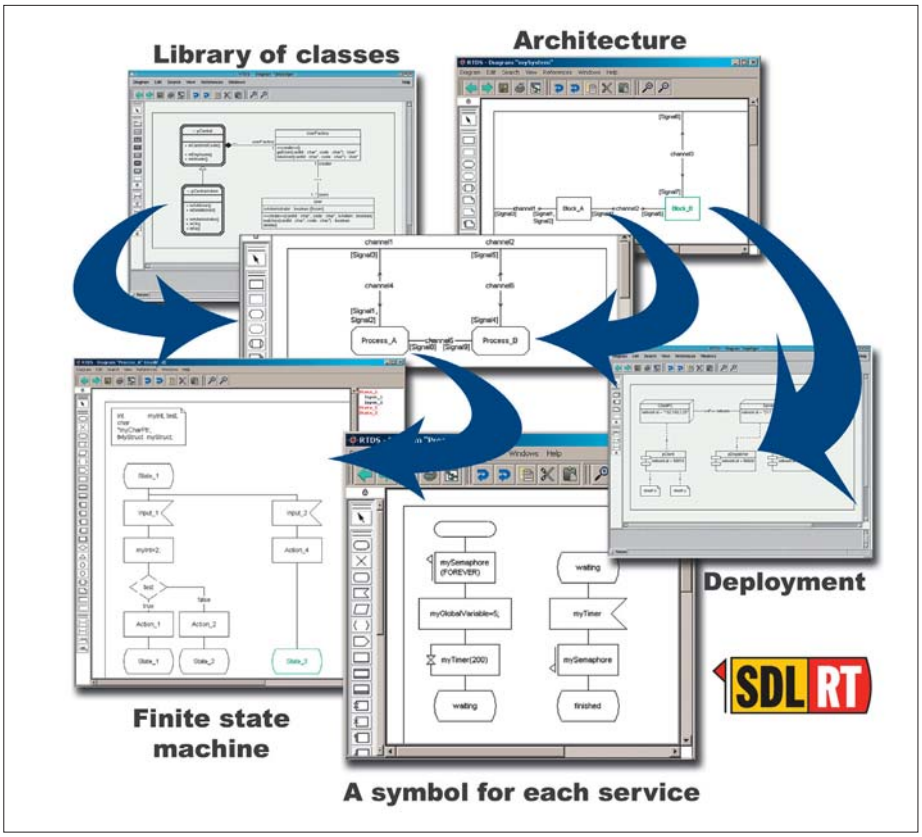
NI CompactRIO Embedded Hardware Platform

When the LEGO Group needed parallel programming and motor control tools intuitive enough for children, it selected graphical software powered by NI LabVIEW. With LabVIEW graphical system design, domain experts can quickly develop complex, embedded real-time systems with FPGAs, DSPs, and microprocessors.

>> Expand your imagination with technical resources at **ni.com/imagine**

Austria 0662 / 45 79 90 0
Belgium 02/757 00 20
Czech Republic 420 224 235 774
Denmark 45 45 76 26 00
Finland (09) 725 725 11
France 01 57 66 24 24
Germany 089 / 741 31 30
Hungary 36 23 501 580
Ireland 353 (0) 1867 4374
Israel 03 6393737
Italy 39 02 41309277
Netherlands (0348) 433466
Norway 47 (0) 66 90 76 60
Poland 48 22 3390 150
Portugal 351 210 311 210
Russia 7 495 783 6851
Slovenia/Croatia 386 3 425 4200
Spain (91) 640 0085
Sweden 08 587 895 00
Switzerland 056 / 200 51 51
UK 01635 523545

NATIONAL INSTRUMENTS™



SDL-RT diagrams

- most of the tasks are based on extended finite state machines,
- tasks communicate with messages sent to message queues,
- built-in timers help to deal with unexpected behaviour.

Most telecommunications manufacturers have used SDL to design their software and some of them have measured the added value when using this modeling language. The results are impressive: it has increased quality by a ratio of 5 and reduced the overall development time by 35% in average.

In 1997, the object management group standardised the Unified Modelling Language, a merge of different object-oriented modelling approaches, most of them coming from the database application domain. The first versions of UML, versions 1.x, were too generic to support a platform-independent model. They did not contain any of the real-time operating system concepts such as task, semaphores, and timers. They did not contain any semantic, nor any data type. In order to use UML to describe a PIM, version 2 of UML introduced the concept of profiles to make UML more precise within each application domain. A profile allows to introduce specialized concepts and some semantic within a UML model. But the OMG did not define any standardised profile. Therefore UML 2 tools have introduced their own profiles, most of the time without documenting them, making the models tied to the tools they have been designed with, and tied to the underlying profile that was used. The ITU has taken this opportunity to work on a UML profile for telecommunications systems based on SDL under the Z.109 reference. The standard should be in force by the end of 2007, at that time SDL will be a standardised UML profile for telecommunications systems.

Because UML is very abstract and informal it is mostly used in the early phases of the development process when analysing and setting the requirements on the system. When it comes to coding, traditional textual languages are at the same level as the SDL abstract data types. Because of its graphical abstractions dedicated to telecommunication systems, SDL is positioned between the very generic UML and the very specialized coding languages.

When using SDL, telecommunications manufacturers found the concepts within the language were not exactly the ones available in the real-time operating systems nor the ones in textual languages. For example, in SDL messages have priorities and processes do not; on an RTOS it is the other way around, tasks have priorities but messages do not. As another example, in SDL it is possible to define arrays indexed on reals, which is very tricky to implement in C. To fully conform to SDL, code generators had to support all the SDL concepts and produced complex and illegible code. So in order to be efficient, most of the telecommunications manufacturers broke the SDL semantic to use the one from their operating systems, and wrote C code manipulating C data types instead of the SDL data and syntax. They were actually already using what we call SDL-RT. As previously explained, SDL-RT comes from industrial practice and its first version has been written by PragmaDev so that everybody has a common way of combining UML, SDL, and C or C++. The specification is freely available on http://www.sdl-rt.otg, and is easy to read compared to an official standard. The storage format is XML so that the description is not tied to any tool. Since SDL-RT aims at all real-time and embedded applications, the semaphore concept has been introduced in the language so that each service of a real-time operating system has a dedicated graphical symbol. SDL-RT concepts are now proposed to ITU in order to be standardised. So eventually SDL-RT will most likely become a standardised UML profile for real-time applications.

SDL-RT diagrams are:

- the class diagram in which passive classes are further described in C++ and active classes are further described using the SDL finite state machine,
- the architecture diagram which decomposes the system in hierarchical functional blocks down to the process level,
- the process behaviour which is described using the SDL finite state machine because it is much more detailed than ones from UML,
- the deployment diagram for distributed systems,
- the message sequence chart (MSC) to document execution scenarios similar to the UML sequence diagram.

Each service of the RTOS has a dedicated symbol: dynamic task creation or deletion, message input and output, timer start, cancel, and time out, semaphore takes or gives.. All views are related to each other making the model of the system consistent.

Since its inception twenty years ago, SDL has been a model-driven language for the specification of telecommunications systems. Following industrial practice, SDL-RT has extended the application domain to all applications based on real-time operating systems.. In the meantime, UML 1 being too generic, UML 2 has introduced the possibility to define profiles dedicated to an application domain but has not standardised any. Each UML 2 tool has therefore implemented a proprietary profile that is rarely documented making portability to another tool impossible. The ITU is standardising a UML profile for telecommunications systems based on SDL. The profile should be finalised by the end of 2007. ■

Steer-by-wire system development with model-based design

By Joachim Langenwalter and Tom Erkkinen, The MathWorks

Model-based design enables the automatic generation of final-build software from models for high-volume automotive embedded systems. A software engineering framework is needed to support this. This article presents a framework of processes, methods and tools for the design of automotive embedded systems.

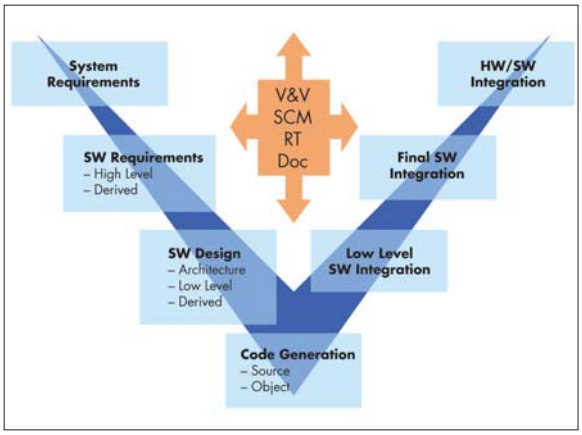


Figure 1. V diagram of the software process

■ Production code successes have recently been reported in various industries from companies including Denso, Motorola and Toyota. This technology is taking hold as an important component within the next evolution of software development. Although the impact on the software engineering process as a whole is understood, it has not been clearly established. This is especially apparent to participants of previous, similar evolutions including the migration from machine code to assembly code, and then to source code. With increased abstraction and automation came new processes, methods and tools.

However, due to difficulty of use, understanding or limited tool support, not every good idea has flourished. Evidence shows that these methods and tools are not always practical for mainstream production usage. For example, formal methods where proofs are used to ensure software correctness are written in a language that only a few experts worldwide truly understand. Further, real-time Case tools in the 1980s aided design, but did not provide an easy path to final code. Production code generation has done well in the early adoption stage, due mainly to its practicality. However, further growth requires integrated process, methods and tools to support it. A new process will be successful only with such methods and tools. If one of these pieces is missing, the effort to re-engineer a company's mature embedded system is no longer feasible or practical. This article presents such a framework focused on production code generation:

- Process – model-based design.
- Methods – modelling, simulation, rapid prototyping, production code generation, model testing and coverage, and in-the-loop testing.
- Tools – development tools, V&V tools, and integral tools.

Model-based design supports the needs of controls/DSP systems engineers and software developers by providing a common environment for graphical specification and analysis. In this process, models are made and used to specify system data, interfaces, feedback control logic, discrete/state logic, and real-time behaviour. Model-based design is used in nearly every industry that requires embedded control systems development. It is particularly well-entrenched in development processes for embedded applications such as large-scale automotive electronic control units. DSP and communications applications also use this approach, but emphasise modelling and prototyping rather than production code generation. To satisfy these various applications, the model-based design process must address the needs of safety-critical systems such as steer-by-wire systems. The process must also yield a final, executable code that is extremely compact, fast and traceable. This is due to the high volume nature of mass-produced ECUs, which necessitate the use of low-cost, fixed point microcontroller units and DSPs.

Model-based design fits within the context of any process framework, including those characterised in IEEE software engineering standards. IEEE Std. 730 applies to any general-purpose software project. A good understanding of its process framework results from a review of its stated requirements for critical project documentation.

Model-based design places great emphasis on process iterations, early testing, and reuse throughout the development process, making it both unique and powerful. The practicality inherent to this process is demonstrated at the bottom of the V: production code generation is an automatic transition from design. In model-based design, a block diagram or a state diagram model can serve as the system and software requirements, software design, or, with a slight change of perceptions, the source code. Also unique to this process is the extensive V&V effort that is made prior to a final build.

The benefit of early V&V is clear: fewer bugs will be found and less rework will be performed during final system integration and test. Catching a bug on the desktop is highly preferable to

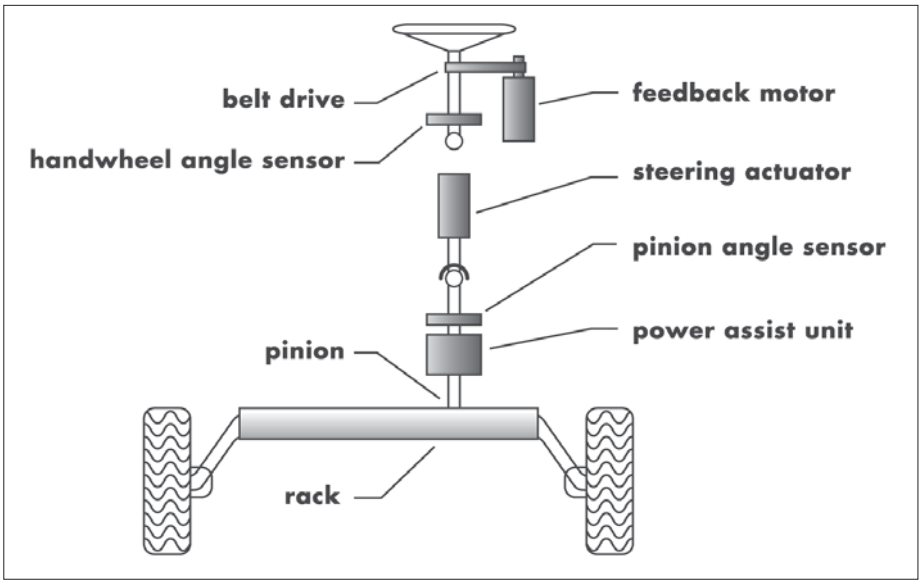


Figure 2. Steer-by-wire system

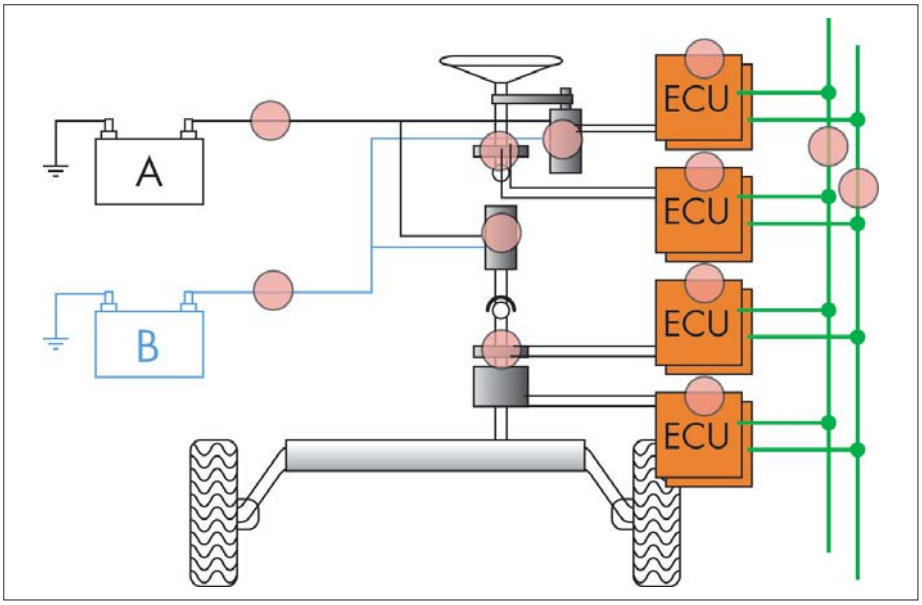


Figure 3. Possible faults on the steer-by-wire system

encountering it during a winter test drive in Finland. Organisations can leverage this benefit to achieve faster time to market.

Model-based design methods are employed during the software engineering process. A concise description of each development method is provided below, with examples and tool support information. Note that all the tools shown herein are commercially available. The following sections step through the development activity and include key V&V methods. Finally, the article concludes with the integral components.

Models are used for specifying requirements and design for all aspects of every individual subsystem (e.g. steer-by-wire). A typical system includes: inputs (e.g., steering wheel sensors), controller or DSP model, plant model (DC motor, rack and pinion, wheels), outputs (change of direction). As shown, a system model can be created to represent the desired behaviour using control system block diagrams for feedback control, state machines for discrete events and conditional logic, and DSP blocks for filters. The model is then executed and analysed to ensure that the requirements are satisfied, using methods such as time- or event-based simulation and frequency domain analysis. For example, a steer-by-wire system must respond to a sensor failure and "shall attenuate high frequency response below 3db and not lag commanded rate by more than 1.5 m/sec".

Modelling and simulation of the steer-by-wire system in figure 2 determine if these requirements conflict or are valid. Simulation is a core validation activity and ensures that a system can be realised to satisfy the requirements. Due to inaccurate plant models and insufficient processing power required to get a working solution on production silicon, modelling alone does not provide the total solution. To overcome these shortcomings, rapid prototyping is extremely useful because it replaces the plant model with the physical plant. In the steer-by-wire example the plant might be a car, and in that case, an actual car is used. However, because the system is not yet built, a real-time or embedded platform runs the controller software and interacts with the plant.

There are two forms of rapid prototyping: functional and on-target. Functional prototyping uses a powerful real-time computer such as a multiprocessor floating point PowerPC or DSP system. The goal is to determine if the system controls the physical car as well as it controlled the modelled car. If so, the plant model inaccuracies are shown to be insignificant, and the control strategy is validated. On-target rapid prototyping executes the software in the same or similar production MCU or DSP, rather than a high-end PowerPC core or other dedicated high-end rapid prototyping hardware.

The goal is to download the code into the actual production target for quick testing with the physical plant. If it performs well, the controller is not only deemed to be valid, it can feasibly be realised in production. The software design activity includes fixed-point data specification, real-time tasking, data typing, built-in-test and diagnostics. With model-based design, the same model used for algorithm specification and validation is refined and constrained by the software engineers as part of the production code generation process.

It is more beneficial to test the model on a desktop rather than deploy it on hardware for build and integration. Testing based on source code has existed for many years, but recent methods now allow for model testing and structural coverage. The usage scenario is that a developer fully stresses the controller to verify its design integrity using simulation and coverage. Another type of testing is failure mode effect analysis, to ensure the safe operation of the steer-by-wire under fault condition, as shown in figure 3. Examples of poor design integrity are numerical overflow and/or dead code. Stress testing of the model using minimum and maximum numerical values helps ensure that overflow conditions will not occur. This type of testing is easy with simulation, but dead code is not so easily detected because detection requires structural coverage. Dead code differs from deactivated code in that the latter is known to the developer and is deactivated for a reason. Actual dead code means something was missed during specification.

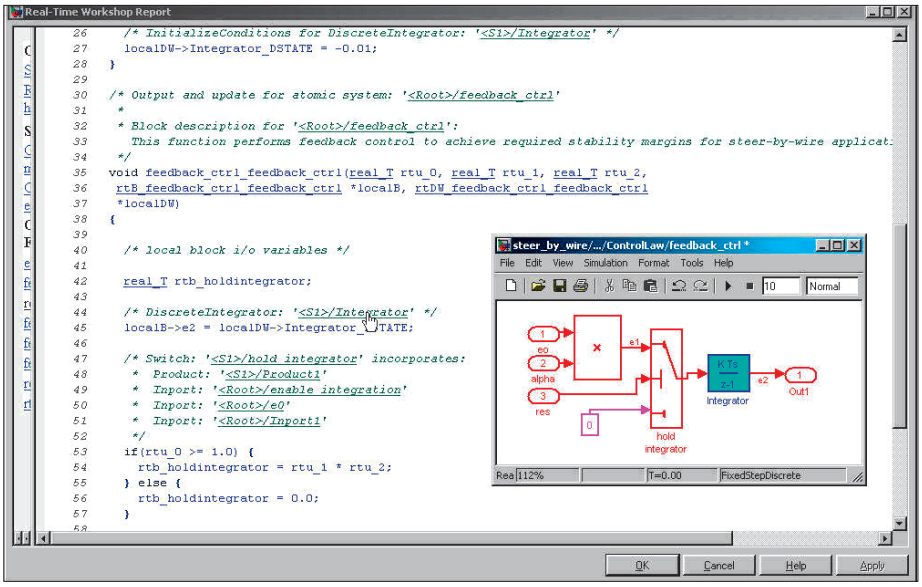


Figure 4. Code review

Model coverage assesses the cumulative results of a test suite to determine which blocks were not executed or which states were not reached. Certain types of coverages are well established in source code languages (such as C, C++) but coverages are now available for the model. This effort required new theory and tools not needed (or possible) for C, since these languages do not posses constructs such as blocks or states.

Modified condition/decision coverage (MC/DC) is considered by the FAA to be the most stringent coverage level necessary to satisfy safety critical systems. This coverage, among others, is now available within a model-based design framework and is in many cases also required for X-by-wire designs. Modern embedded systems contain several distributed ECUs, which communicate in real-time with each other over a fault-tolerant communication system like FlexRay.

The latest DSC (dynamic stability control) of BMW contains ABS as one of 15 sub-functionalities. By adding blocks of networks components such as hosts, tasks, signals, etc from DECOMSYS to the individual subsystems, the embedded functions can be connected and mapped onto an architecture of ECUs.

Furthermore, the simulation of the temporal behaviour of task activations of a time-triggered operating system such as OSEKtime/OS is facilitated. Clusters, hosts, tasks, and connections are designed and simulated within the MATLAB/Simulink environment. Finally the whole design integrates seamlessly with the DECOMSYS DESIGNER product for interaction with the FlexRay xCDEF design data repository. Whereas the distributed network design solutions from Vector (DaVinci) and Cadence (SysDesign) integrate the RTW/RTW-EC from SL/SF (Simulink/Stateflow) generated code from subsystems and map them onto the architectures for verification. Now that the model has been verified and validated, it is time to generate code. As with a compiler, this process is straightforward. Various optimisation settings and user configuration options exist. The key is to keep the code efficient, accurate and integrated with legacy code or other tools. It is also important for the code to be traceable to the diagram, so that it may be reviewed and verified.

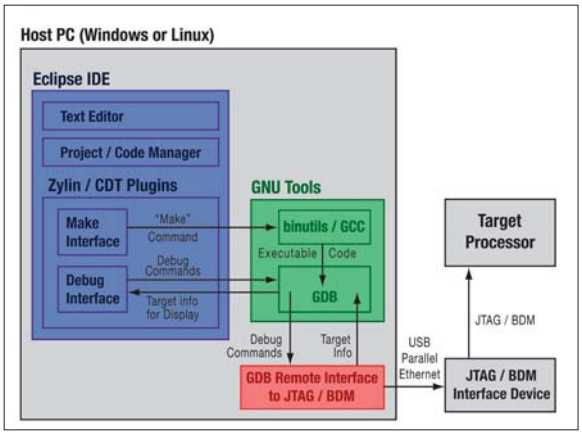
Figure 4 shows an automatically linked HTML report. As shown, when the developer selects the sum block in the code, it highlights the sum block in the diagram. Figure 4 uses rate transition blocks, however direct links also exist to commercial RTOSs, including VxWorks and OSEK variants. As shown in the example below, device driver integration is also needed.

Once the controller has been built, a series of open- and closed-loop tests can be performed with the real-time plant model in the loop. One example involves only the processor, and is known as "processor-in-the-loop" testing. The second example uses the actual built ECU hardware, termed "hardware-in-the-loop". In either case, the physical controller is tested with the plant model. Through a series of tests, perhaps the same test used during requirements validation, the controller must be shown as acceptable to the customer. Most software standards require traceability of requirements, perhaps originating in other requirements tools, throughout development. Also, software configuration management (SCM) is needed to store, version and retrieve the various development artefacts. Documentation via report generators ensures management, customers and suppliers will see the model. ■

Building an embedded cross-development environment with Eclipse

By Brian Handley, Macraigor Systems

This article describes how to construct a free or low-cost cross-development environment based on the open-source Eclipse IDE and GNU toolsets. The task can be difficult and time-consuming, but the article shows how preconfigured Eclipse projects and pre-built GNU tools help speed construction.



Development system using a JTAG/BDM target connection

The Eclipse development environment has become the de-facto industry-standard environment in which to host embedded development tools. Many of the traditional embedded tools vendors who used to sell their own proprietary development tools and environments have embraced Eclipse, and ported their products to run within it to take advantage of the sophisticated, feature-rich framework provided by the Eclipse IDE. These tools and environments are powerful, but they can still be expensive. For projects on a tight budget, it is now possible to use the freely available, open-source Eclipse IDE along with the open-source GNU tools (binutils, gcc and gdb) to construct a complete cross-development environment at little or no cost.

However, piecing together all the necessary components to build a system such as this is not necessarily easy. Eclipse was not originally built to handle cross-development, or even the C or C++ languages typically used in most embedded projects. Therefore, a significant amount of effort is needed to get Eclipse to perform this task adequately. Beyond this, Eclipse does not currently have any concept of a remote debug connection on its own. If a debug interface such as JTAG or BDM, or even an Ethernet or serial connection to a target-resident debug monitor, is used, Eclipse must be reconfigured to handle this situation.

Additionally, the required GNU tools are typically available only in source format, and they must be built for the particular host and target processor being used by a project. Getting these tools to build for a particular host-target combination can be difficult, consuming engineering time that could be better spent on application development. This article describes how to construct a free or low-cost cross-development environment based on the open-source Eclipse IDE and GNU toolsets. The task can be difficult and time-consuming, and this article also presents the preconfigured Eclipse projects and prebuilt GNU tools offered by Macraigor Systems to help speed construction.

To build a functional, free, cross-development environment, several components must be obtained and integrated together. The Eclipse development environment is the framework into which the other necessary tools are integrated. Eclipse itself includes an editor, project manager and debugger interface. Since the environment is intended for embedded cross-development, the C and C++ languages must be supported. This requires use of the CDT (http://www.eclipse.org/cdt/) plug-in for Eclipse. The assembler, compiler, linker and other code-generation utilities will be provided by open-source GNU code. If the goal were to develop native applications in C/C++ using Eclipse, then these tools would suffice. However, for embedded cross-development, a few more pieces are needed. Eclipse with the CDT plug-in has no concept of using a remote debug connection in order to connect to an embedded processor. Zylin AS Consulting offers the open-source Embedded CDT and another plug-in, which together allow the Eclipse debugger to connect to a remote target via any debug connection (see reference to Zylin plug-ins below). This debug connection is typically a JTAG, BDM, Ethernet or serial connection. In addition, if a JTAG or BDM connection to the target is needed, a method must be provided for the GNU Project Debugger (GDB) to communicate to the target using these interfaces. The completed development system using a JTAG/BDM target connection is shown in the figure. A discussion of each of the above required components follows.

According to the official Eclipse website, the Eclipse Foundation manages open-source development of "projects which are focused on providing a vendor-neutral open development platform and application frameworks for building software". The Eclipse Foundation has created the Eclipse platform, which provides a feature-rich integrated development environment with a well defined interface that allows additional features to plug in and work seamlessly with the existing code. Eclipse has rapidly gained favour among embedded tools compa-

Size it up and scale it to fit

The 3-in-1 evaluation platform XC800 USCALE Start Kit

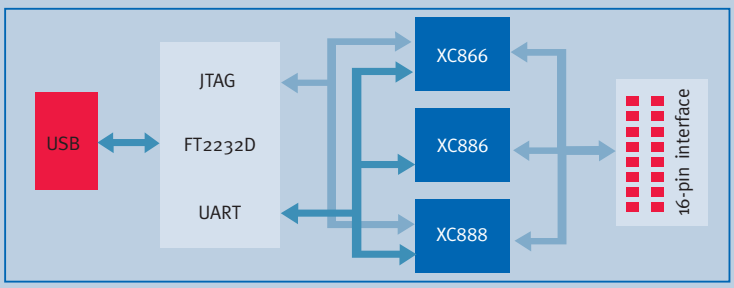
THE XC800 USCALE START KIT is a remarkably affordable tool for evaluating the XC866, XC886 and XC888 family of microcontrollers. This low cost kit provides a simple way to evaluate the performance and feature scalability of our XC800 family of high-performance 8-bit microcontrollers on a single convenient USB platform. Simply install the software and plug the stick into a USB port to access the full functionality of a powerful evaluation tool, enabling you to bring the competitive benefits of a high-performance 8051 core with on-chip Flash and a powerful mix of peripherals such as an enhanced CAPCOM6 (CC6), CAN, LIN and 10-bit ADC to your future products.

Key features of XC800 products

- High-performance 8-bit 8051
- Embedded Flash
- Powerful peripheral set
- On-chip OSC and EVR
- Full tool-chain support
- Industrial and automotive quality

Key applications

- Motor control
- Washing machines
- Transportation
- Building automation
- Air conditioners
- Industrial automation



Size it up and scale it to fit with the XC800 USCALE START KIT.

Order now at:

www.infineon.com/XC800USCALE

Never stop thinking

nies because it provides a sophisticated IDE into which they can plug their tools and no longer have to worry about building and maintaining their own proprietary environments. In addition, the Eclipse public license allows the creator of derivative works based on Eclipse to retain their distribution rights, so companies can focus on their core embedded competencies and still profit from their efforts. This is excellent news for those trying to put together a free development environment. It makes available a commercial-quality IDE that has the backing and support of a large number of both embedded and enterprise-software tools companies. However, as mentioned earlier, Eclipse by itself provides only a framework and some generic tools, such as an editor, code/project manager and debugger interface. In order to construct a cross-development system, several more packages must be obtained and integrated into Eclipse.

Eclipse was originally developed in and for the Java programming language, and the basic framework is still specific to the Java language. Most embedded cross-development projects still have device drivers, operating system (OS) code and applications written in C, C++ or assembly language. To make Eclipse compatible with, and useable for, C/C++, a subproject called C/C++ Development Tooling was created to build a plug-in that would add these features to the basic Eclipse framework. This plug-in is available as a free download from Eclipse at http://www.eclipse.org/cdt/downloads.php.

With the Eclipse framework and CDT in place, the environment is capable of supporting and enabling code development in C/C++. However this environment will only work for native application development. For embedded cross-development, there are still some issues that must be addressed, primarily the handling of remote debug connections to a target processor. This problem is discussed further in the context of Zylin plug-ins. In addition, an assembler, compiler, linker and loader are still required for the specific target processor that will be used on the project. The next section discusses using GNU tools that provide these utilities.

The Free Software Foundation makes available free source code for a wide range of programs and utilities, including a set of tools that together can provide everything necessary to build, link, load and debug an embedded application. The combination of GNU binutils (http://www.gnu.org/software/binutils/), the GNU compiler collection (GCC) (http://gcc.gnu.org/) and the GDB (http://sources.redhat.com/gdb/) provides a fairly complete toolset for building and debugging embedded applications. These tools can be used on their own for this purpose. The binutils package provides an assembler, linker, archiver and several other utilities for code development, the GCC provides the C/C++ compiler, and the GDB allows the code to be downloaded to and debugged on the target processor. The downside of using these tools as they are is that there is no graphical user interface (GUI) and no real integration of the tools. Used by themselves, the tools basically provide a command-line interface. However, Eclipse with the CDT is capable of sufficiently integrating these tools into an environment with a GUI so most of the command-line use of the tools can be avoided. The other problem with using the GNU tools for code development is that these tools are generally provided only in source form. Although the tools support a huge array of various target processors and just about any host OS and hardware one could imagine, the user usually has to configure and build the tools. This build process can be a time-consuming, frustrating experience, especially for someone who has not done it before. Fortunately, Macraigor Systems provides an alternative to the labour-intensive task of building GNU tools in-house. This is described in more detail below.

An environment consisting of Eclipse, the CDT plug-in and the appropriate GNU tools is close to being a functional embedded cross-development system. As mentioned above, however, Eclipse and the CDT do not support remote target connections to an embedded processor. They assume that debugging is occurring on the host machine. In order to download the embedded code from the host to the target and then connect to a debug agent of some type running on the target hardware, some changes must be made in the way the CDT handles debugging. To address this problem, Zylin AS Consulting, has created and made available Embedded CDT and another small plug-in that together "understand" and properly handle embedded debugging using the GDB from within Eclipse. These free plug-ins are available at http://www.zylin.com/embeddedcdt.html.

An Eclipse Project called the device software development platform (DSDP) is now available. This project is specifically aimed at enabling Eclipse to be used for embedded cross-development so that, at some point in the future, the Zylin Embedded CDT modifications may become unnecessary. Further information about the DSDP project can be found at http://www.eclipse.org/dsdp/. The only missing piece that remains in the integrated cross-development system is some type of debug communication method to connect the host computer to the target processor. Traditionally, this connection is usually made via a serial, Ethernet or JTAG/BDM interface. If a project is using hardware for which a board support package already exists, it may be feasible to simply run a GDB debug agent on the target and connect the GDB to it using a serial or Ethernet connection. However, for new custom target boards, the interface of choice is usually JTAG or BDM. These types of debug interfaces are built into most of the more popular embedded processors including ARM, MIPS, PowerPC and XScale processors, as well as many others. These interfaces provide a dedicated debug connection directly to the target processor that has several advantages over using a serial or Ethernet connection:

- They are built into the processor and typically only require that the processor is
- They can be used to write and debug boot code and drivers.
- They do not use any valuable target resources.

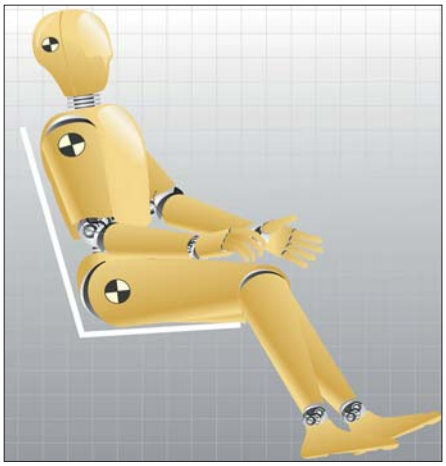
To connect a JTAG or BDM interface device to the target using the Eclipse/GNU environment described above, a debug agent of some sort is required. The GDB has a well-defined back-end interface called GDB Remote that has become a common standard for connecting the debugger to an embedded processor via a JTAG or BDM connection. This is usually handled by a proprietary stand-alone utility that runs on the host and provides a TCP/IP port that will accept a GDB connection on the front end and connect to the JTAG/BDM hardware device on the back end. Most vendors of JTAG/BDM interface devices provide a utility for this type of connection with their hardware. By using readily available, free, open-source software tools, it is possible to construct a full-featured, integrated environment for embedded cross-development. The process of gathering components, integrating them, and in the case of the GNU tools building applications from source, is probably well within the capabilities of most embedded software engineers. However, this process can still be a time-consuming and challenging task.

Macraigor Systems has simplified the IDE construction described in this article by offering free downloads of a suite of tools that includes Eclipse, the Zylin Embedded CDT plug-ins, pre-built GNU tool kits for AMDx86, ARM, MIPS, PowerPC, XScale, ColdFire/CPU32, and Freescale DSP target processors and more than 70 example Eclipse projects configured for standard evaluation boards using various embedded processors. Also available is a document containing detailed instructions on downloading, installing and testing the environment with actual target hardware. The preconfigured Eclipse projects and the pre-built GNU tools, with install programs for Windows OSs and RPM scripts for Linux OSs, allow a user to quickly get the complete environment up and running on actual hardware. ■

Solid project lifecycle planning with IEC 61508 software development

By Richard Barry, Wittenstein high integrity systems

This article provides a basic overview of IEC 61508, and highlights the sections of the standard that are specific to software development.



IEC 61508 is an international standard covering the development and usage of electrical, electronic and programmable electronic, safety-related systems. In this context, a safety related system is a system that performs one or more safety functions. IEC 61508 covers both hardware and software development, so, with due consideration, safety functions can be performed by software. IEC 61508 is widely used in the industrial automation and control fields, and also forms the basis for a growing selection of safety standards that have been tailored to meet the needs of specific industry sectors. Examples of such tailored standards include IEC 61513 for the nuclear power industry, IEC 61800-5-2 for power drive systems, and IEC 61511 for the process industry.

The standard defines the analysis, design, implementation, production and test requirements for safety-related systems in accordance to the Safety Integrity Level (SIL) assigned to the system. The SIL is assigned according to the risks associated with the use of the system under development, with a maximum SIL of 4 being assigned to systems with the highest perceived risk. The higher the assigned SIL number the lower the rate of failure must be for all identified unsafe failure modes. IEC 61508 comprise seven parts. Figure 1 illustrates the relationship between these parts. Parts 1 to 3 of the standard contain the primary information, with supplementary material provided by parts 4 to 7. As a whole, the seven parts define a system development Safety Lifecycle. The first three parts can be briefly summarised as:

✓ Part 1 defines the necessary development management system, including how the system safety requirements are to be calculated, elicited and defined.

✓ Part 2 relates to the hardware aspects of the system development. It contains the techniques required to reduce both systematic and random hardware failures.

✓ Part 3 relates to the software aspects of the system development. Software cannot suffer random failures, and therefore part 3 contains the techniques required to guard against systematic failures.

Systematic failures are generally not quantifiable, thus part 3 compliance provides a confidence level rather than a calculable failure rate. So who is to say whether or not the confidence level has been reached? To credibly claim compliance with IEC 61508 it is necessary to have your development processes and compliance evidence assessed by a third party who must be accepted as an expert in the field. While there is no body with a legal mandate to issue a de facto compliance certificate, there are several companies which can, as a commercial service, provide an audit and then offer a statement of your compliance (or not as the case may be). A functional safety assessment, and the associated SIL assignment, is applied to a system as a whole. Certification at the component level, whether software or hardware, to a specified SIL does not indicate that the component can be used without regard in a system that has been assigned a SIL of the same value or lower. Any certification evidence supplied by the component provider can only be used as evidence in the context of risks highlighted during the safety assessment of the entire system. The use of 'certified' components does not avoid the requirement to perform a safety assessment on the entire system. Certified components should, however, provide the system builder with a lower cost, faster, and much lower risk route to IEC 61508 compliance.

The techniques and measures described by part 2 with reference to the calculation and reduction of the hardware failure rate are mature and largely well understood. The same is not necessarily true of the part 3 software-related requirements. IEC 61508 is orientated around the 'V' development model, a simplistic version of which is depicted in figure 2. While essentially a top-down model, there are plenty of feedback paths permitting a more pragmatic iterative approach, provided the approach taken follows

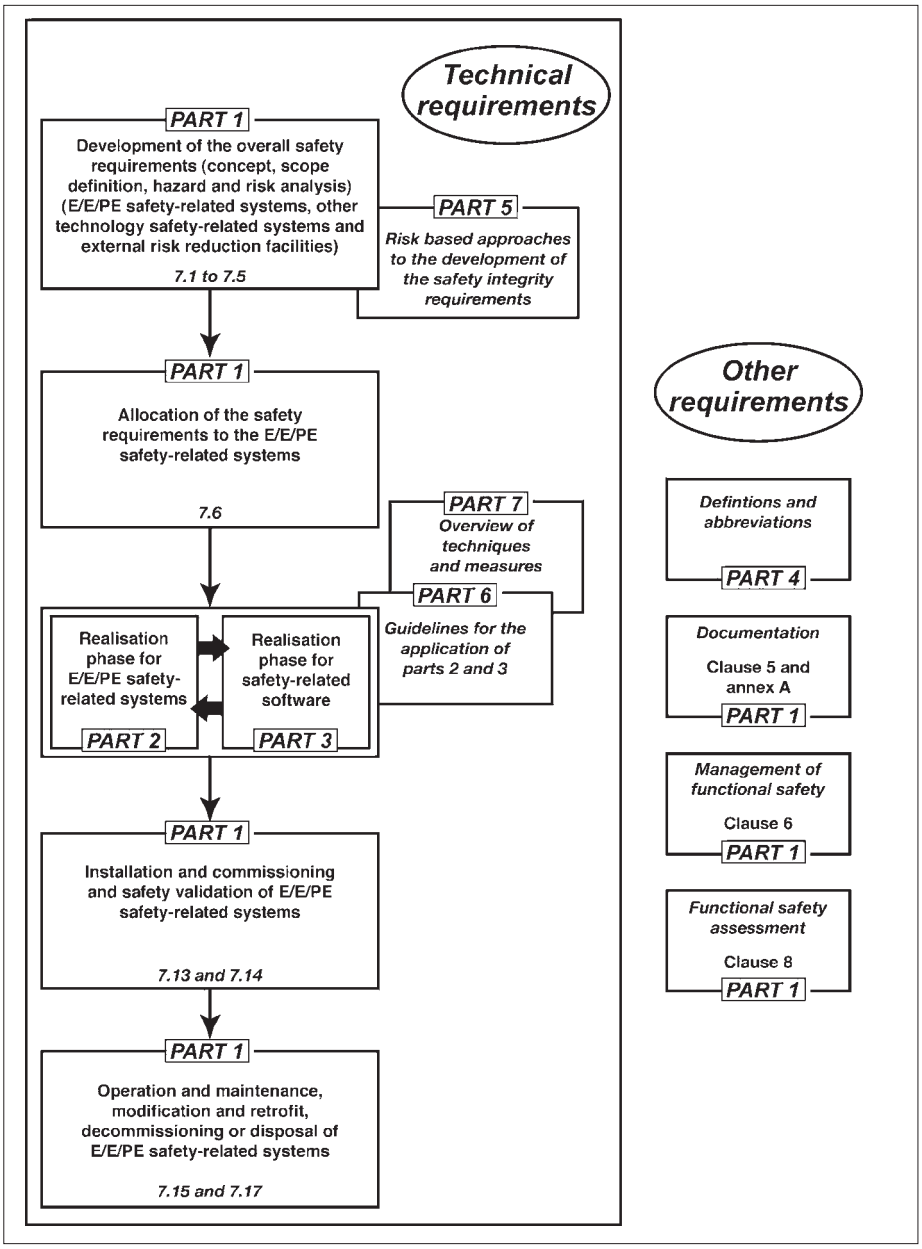


Figure 1. Illustration of the relationship between the seven parts of IEC 61508

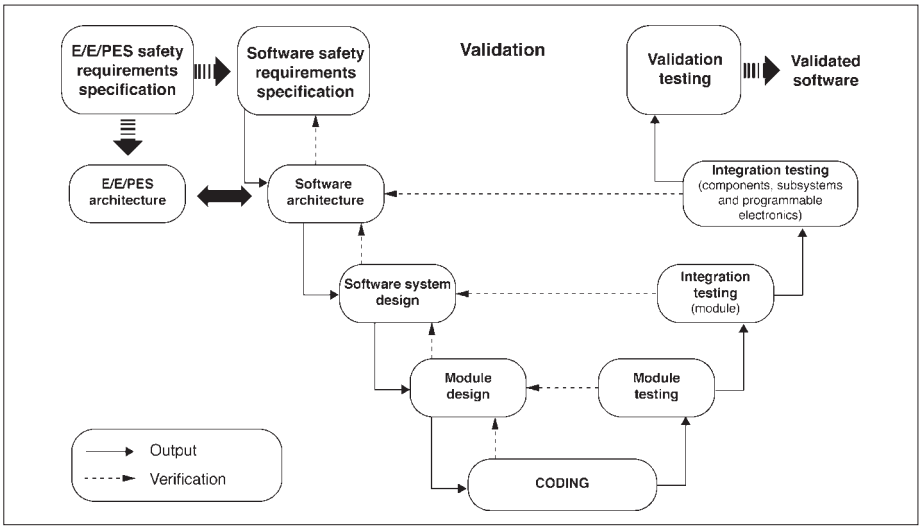


Figure 2. IEC 61508 is orientated around the 'V' development model

the development plans submitted to the third party assessor. The primary path through the V is to specify and develop the design along the downward path of the V whilst simultaneously locking in the verification definition used during the upward leg of the V. The output of the engineering process is documentation and the V dictates the sequence and synchronisation of the documents.

Annex A to part 3 provides ten tables to assist in the selection of techniques and measures to be undertaken during the phases of the V development process. Annex B provides additional and finer detailed information, also in tabular form. The table headings are listed in the sidebar. As can be seen, tables identify items that would typically be expected for any safety-related software development, although the table entry 'Support tools and programming language' is noteworthy and given special consideration later in this article. Both the Annex A and Annex B tables contain the six columns depicted in figure 3.

The 'Technique/Measure' column is self explanatory, other than to note that letters following a row number are used to indicate an option. For example, if there is a row 3a and a row 3b, then the developer need only demonstrate compliance with the content of either row rather than with both. The 'Ref' column simply provides information on where further information or clarification on the technique or measure can be sourced within the IEC 61508 document itself. The 'SIL1', 'SIL2', 'SIL3' and 'SIL4' columns are used to indicate what is required for each of the respective levels. Entered in each of these columns will be one of the four possible values described in the table below.

Entry	Meaning
NR	The described technique or measure is Not Recommended for the given SIL. Its use at that SIL is actually discouraged.
---	There is neither a recommendation for or against the use of the technique or measure.
R	The use of the technique or measure is Recommended.
HR	The use of the technique or measure is Highly Recommended. Not to use the technique or measure therefore requires the formulation of a rationale for its omission to be submitted to, and accepted by, the conformance assessor during the safety planning process.

As would be expected, the higher the SIL, the higher the number of HR entries encountered. IEC 61508 is not a lightweight standard and a systematic approach is required in order to form a cost-effective compliance argument. Compliance with the items denoted as Highly Recommended (HR) within the annex A and B tables can be unambiguously achieved by extending the tabular format to include a column that cross references each Highly Recommended technique or measure to the appropriate paragraph within the system development planning documentation. Unfortunately not all the IEC 61508 requirements are contained within tables. Sentences within the text that contain the word 'shall' also require compliance consideration.

An easy if rather laborious method we have successfully used to demonstrate compliance to these textual requirements is to extract each 'shall statement' into a separate compliance matrix, which is then completed during the project planning stages. The need for solid project lifecycle planning has already been highlighted several times. It is not only essential to minimise the commercial risk of the project, but is in fact a requirement of the standard itself. When seeking third party assessment it is essential to submit the development plans to the assessor prior to the commencement of any formal development activities.

Safety-related developments are by their nature costly so pre-approval of the processes to be undertaken prior to commencement can prevent expensive mistakes. It would be expected for a software project development to a high SIL level to include as a minimum a planning documentation (system/safety development plan, software development plan, requirements management plan, configuration management plan, software test plan, standards compliance statement/compliance matrix). IEC 61508 places requirements on the programming language and compiler used. For example, for projects with a high SIL value, the programming language is Highly Recommended to be strongly typed, and the compiler Highly Recommended to either be certified itself, or alternatively have 'proof in use' evidence of its correctness available.

Fundamentally we need to demonstrate that our system is safe, so any evidence chosen to demonstrate the correctness of the compiler through proof in use needs to be bullet-proof. It can be argued that the assembly of such evidence requires information on: the number of

Sidebar – Names of the relevant Part 3 tables

The 10 Annex A tables are titled:

1. Software safety requirements specification,
2. Software design and development: Software architecture,
3. Software design and development: Support tools and programming language,
4. Software design and development: Detailed design,
5. Software design and development: Software module testing and integration,
6. Programmable electronics integration (hardware and software),
7. Software safety validation,
8. Modification,
9. Software verification, and
10. Functional safety assessment.

Annex B tables are titled:

1. Design and coding standards,
2. Dynamic analysis and testing,
3. Functional and black box testing,
4. Failure analysis,
5. Modelling,
6. Performance testing,
7. Semi-formal methods,
8. Static analysis.

	Technique/Measure*	Ref.	SIL1	SIL2	SIL3	SIL4
1	Computer-aided specification tools	B.2.4	R	R	HR	HR
2a	Semi-formal methods	Table B.7	R	R	HR	HR
2b	Formal methods including for example, CCS, CSP, HOL, LOTOS, OBJ, temporal logic, VDM and Z	C.2.4	---	R	R	HR

NOTE 1 – The software safety requirements specification will always require a description of the problem in natural language and any necessary mathematical notation that reflects the application.

NOTE 2 – The table reflects additional requirements for specifying the software safety requirements clearly and precisely.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. Only one of the alternate or equivalent techniques/measures has to be satisfied.

Figure 3. The six columns illustrated in the Annex A and Annex B table

users of the compiler; what these users are compiling and how often and over what period (with exactly the same compiler version); what bugs have been discovered in the compiler; and assurance that all discovered bugs have been reported. To add to the difficulty, IEC 61508 also requires the utilised tools to be supportable for the lifetime of the product (note product, not project).

Credible conformance to these requirements will not be easy, necessitating special consideration during the project planning phase to ensure any alternative approach undertaken is acceptable to the assessor. An approach we have successfully used to demonstrate the reliance of the compiler (while also not using a strongly typed language) relies on a code coverage metric called unique cause modified condition/decision coverage (MCDC). The premise is that we test the compiler output directly against the system requirements in a manner such that it can be argued that all the compiler output is tested and therefore no reliance is being placed on the compiler itself.

The testing should demonstrate that the compiled code performs all the required functions while not performing any unwanted functions. This method tests the projects output (compiled code) directly against the projects input (requirements), demonstrating that the code output by the compiler conforms to the intention of the requirements themselves, without assuming anything about the behaviour of the compiler itself. This approach may sound laborious, but when working with a formalised requirements and testing process anyway, the extra effort required to demonstrate MCDC need not be great, and has the added benefit of being independent of the compiler used. ■

Product News

Kithara: versatile tool for serial communication

Kithara Software provides with the Serial Toolkit a versatile tool for serial communication under Windows for developers and system dependent programmers. The toolkit supports programmers in developing communication protocols for serial interfaces of the PC. So it enables complex communication solutions for hardware dependent and time critical applications in the industrial automation process.

News ID 341

NI: DIAdem 10.2 supports Windows Vista

National Instruments releases DIAdem 10.2, the latest version of its interactive data management, analysis and reporting software that now offers support for Windows Vista. National Instruments also released NI DataFinder Server Edition 1.1, an updated version of the data mining solution that works with DIAdem 10.2 to simplify data sharing and mining across entire groups or departments.

News ID 259

McObject: embedded database version compatible with MontaVista Linux

McObjects eXtremeDB version 3.1 targets telecom and datacom embedded software with index functions for more efficient communications-related sorting and retrieval, and with an improved high availability sub-system. That makes it a fit with MontaVistas full-featured, deployment-proven Linux CGE 4.0, which adds advanced hard real-time capabilities, new clustering services, and other features for building carrier grade telecom and datacom systems.

News ID 349

MSC: Lattice ispLEVER Classic Software available

MSC announces that Lattice Semiconductor has released a new ispLEVER Classic design tool which can be downloaded free of charge and a one-year, renewable license may be obtained along with the download. IspLEVER Classic now is the complete design environment for CPLDs, GDX/2, XPGAs; ORCAs and FPSCs. All Lattice FPGA families are covered in the full version of ispLEVER which can be purchased at MSC.

News ID 279

Gleichmann: starter kit for NEC V850ES/FX3 MCUs

NEC Electronics' V850FX3-CANIT starter kit was specially developed for the V850ES/FX3 series MCUs and enables to get started quickly with complex CAN applications. The evaluation board, which is available from Gleichmann Electronics, is based on the 100-pin PD70F3377 with 512 KB flash memory.

News ID 287

Telelogic: solution for market-driven product management

Telelogic announces the availability of a new Web-based product management solution for product managers. Based on best practices and existing product management frameworks, Telelogic Focal Point Focus on Product Management provides visualization, prioritization, analysis and planning capabilities to help product development organizations increase the success of their product lines.

News ID 258

BAYER: non-intrusive, hardware-based trace tool

BAYER DSP Solutions announces the new Blackhawk XDS560 Trace System (XDS560 Trace), a non-intrusive, hardware-based trace tool for debugging and profiling high-performance, DSP-based applications providing developers with advanced visibility to debug specialized problems that occur in high-performance, real-time embedded applications.

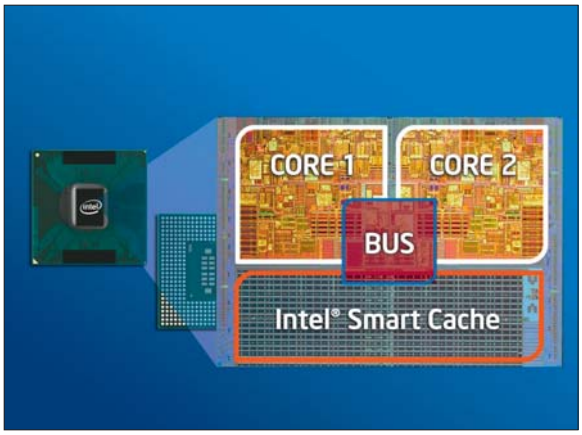
News ID 250

More information about each news story is available on www.embedded-control-europe.com/ece_magazine You just have to type in the "News ID". —

Asymmetric real-time multiprocessing on multi-core CPUs

By Paul Fischer, TenAsys

The latest multi-core processors are ideal for implementing multi-OS embedded applications. Virtualisation technology makes it possible for a multi-core system to easily support multiple operating systems on a single computer platform.



Intel Core microarchitecture processors include multiple CPU cores on a single silicon die.
(Image: Intel)

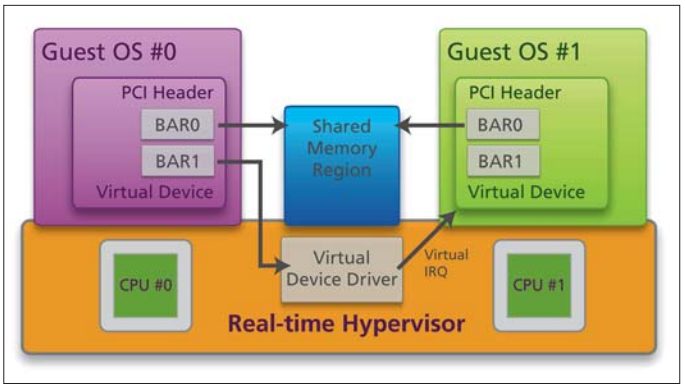
Many real-time embedded applications can be cost-reduced through the use of multi-core processors, integrating applications previously composed of multiple discrete real-time subsystems into single computing platforms with different real-time tasks running on different processor cores. The key to support different real-time OS tasks on different cores is to use a real-time operating system environment that supports virtualisation. Contrary to popular thought, in most hard real-time systems only a small subset of the processor tasks must be capable of delivering hard real-time, or deterministic, performance. By partitioning a software system into subsets that must perform time-critical processing from those which are not time-critical, critical software tasks can be assigned to dedicated hardware resources, ensuring a robust and responsive system design.

Partitioning resources is not a new idea. It is a common design practice. What is new, especially for embedded systems, is the nature of the hardware resources. In the past, an entire processor board or I/O board, such as a dedicated DSP card, would have been allocated for the purpose of real-time data collection, processing, and control. Expensive communication links would have been used to coordinate these stand-alone dedicated real-time hardware resources with high-level supervisory controllers. Now that multi-core processors are available, rather than allocating time-critical software to a dedicated processor or I/O board, embedded system designers can assign a real-time task to a dedicated processor core. In fact, allocating I/O resources to a specific processor core (or cores) on a multi-core processor has the same effect as dedicating a separate processor board, but without the expense of the extra hardware and communications overhead needed to interface the discrete modules.

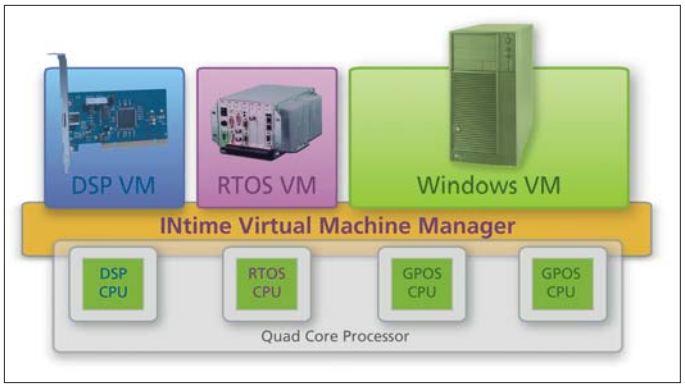
Having CPU cores that are dedicated to executing their own RTOS and real-time applications, with other CPU core(s) hosting a general-purpose OS such as Windows or Linux, the system designer can not only optimise hardware resources, he or she can optimise software engineering resources as well. Such a system is an asymmetric multiprocessing system since the processor cores are not being load-shared across a single OS but are dedicated to specific tasks, in this case simultaneously running multiple operating systems.

Real-time programmers know how to work with low-level hardware and they understand how to tune control systems. User-interface, database, and networking programmers know how to work with high-level APIs and they understand complex data exchange protocols. By giving each discipline the environment they need to get their job done – a real-time operating system for the former and a general purpose operating system for the latter – and hosting both on a single low-cost multi-core platform, one can minimize cost and time-to-market, and maximise the use of one's engineering resources and software technologies. An added advantage is the ability to host legacy real-time applications on separate core (s), minimising software changes to proven code while simultaneously allowing new application code to be hosted on other cores.

Running legacy RTOS code in a virtual real-time machine can also be used to migrate proven real-time applications from obsolete hardware to modern embedded platforms. Because I/O can be virtualised, it is possible to simulate old hardware devices, minimising rewrite of proven legacy code. For example, a VMEbus system could be converted to a less expensive single-board computer system by intercepting I/O requests to legacy VMEbus I/O and redirecting those requests to equivalent on-board I/O devices. To simplify the development of both real-time and human-directed application elements on the same platform, engineers should seek out integrated development environments that support both types of programming. For example, Microsoft Visual Studio can be used to create both Windows applications and real-time tasks that run on TenAsys' INtime OS, within a single solution file, for



Two virtual PCI devices created by the real-time hypervisor are used to implement a shared memory interface between two virtual machines.



With a real-time hypervisor, a quad-core processor can host multiple RTOS and GPOS operating environments.

maximum efficiency. The latest multi-core processors are ideal for implementing multi-OS embedded applications. Virtualisation technology, embedded in the processors, makes it possible for a multi-core system to easily support multiple operating systems on a single computer platform. For example, all Intel Core microarchitecture processors include hardware assistance for implementing virtual hardware systems. A collection of instructions, traps, and a new privileged operating mode is referred to collectively as Intel virtualisation technology, or Intel VT. Intel VT is used by virtual machine manager (VMM) software to host multiple virtual machines on a single hardware platform. Virtualisation technology fills gaps in standard Intel architecture processors making it easier for a VMM to monitor and control access to the supervisory elements of the Intel architecture processor. In other words, virtualisation technology simplifies and assists the process of simultaneously running multiple protected-mode operating systems on a single hardware platform. Virtualisation hardware, such as that integrated into the latest multi-core processors, improves the ability to control access to system level registers, trap interrupts (regardless of masking state), monitor page tables, and control access to the memory used by each operating system on the platform. Virtualisation enables multiple control loops to run simultaneously and improves the ability to isolate I/O and memory, ensuring a distinct boundary between real-time processes and threads from non-deterministic code. Dedicating one core of a multi-core processor to an RTOS and its applications ensures 100% of the CPU instruction cycles of that core are dedicated to real-time processing.

The remaining core(s) are used exclusively by other operating system(s), which could be either a GPOS or another RTOS. Because each processor core is dedicated to an operating system and its tasks, contention for key CPU resources, such as pipelines and the FPU, are avoided, maximising performance and responsiveness of both operating systems. Coordination between the RTOS virtual machine(s) and the GPOS virtual machine(s) is managed by using shared-memory and the built-in inter-processor interrupts, eliminating inter-OS context switch times.

Removing contention for resources in a multi-OS platform has a dramatic impact on real-time performance metrics, such as interrupt latency. TenAsys has measured a 10 to 1 improvement for interrupt latency on dual-core multi-OS platforms compared to equivalent clock speed single-core multi-OS platforms. Latencies measured in the 10-30 microseconds range on single-core systems have been reduced by an order of magnitude to 1-3 microseconds on dual-core systems. With such low latencies real-time application control loops can execute in the 50-200 microsecond range with very high precision, while simultaneously supporting a general purpose OS, such as Windows! An example of an operating environment that supports asymmetric multiprocessing systems is the TenAsys INtime RTOS for Windows.

If you confuse virtual memory with virtual machines, you might conclude that it is not possible to build a deterministic real-time system using virtualisation technology. The two ideas only share a common adjective. One has nothing to do with the other. In fact, in a virtual real-time machine all resources are real and physical: memory, I/O, and CPU cycles. Without this, one could never guarantee the performance or latencies required to call the system hard real-time. Low interrupt latency, direct access to specialized I/O, and a scheduling policy that ensures determinism and priority of real-time functions are key requirements of a real-time virtual machine. Multi-core CPUs and the virtualisation technology built into them are ideal platforms for mixing operating systems on a single system.

By utilising a hybrid approach to allocating resources, compared to conventional virtual machine managers, it is possible to satisfy the needs of two very different environments. In an asymmetric multiprocessing system hosted on a multi-core processor platform, there is no reason to share resources between operating systems. Instead, key physical resources are statically allocated to each virtual machine, based on the needs defined by the embedded system designer. In a conventional VMM access to resources is evenly multiplexed between the virtual machines, in order to maximise use of the machine. This is, however, not a good solution for real-time systems. When determinism and performance are more important than equal access, the VMM must distinguish between resources that must be isolated for use by a specific virtual machine and its guest OS and those I/O resources that are shared between multiple virtual machines.

For example, user interface I/O is usually not associated with time-critical events, so devices like the keyboard, mouse, console, disk, and an enterprise Ethernet interface can be shared between virtual machines. However, hardware that is specific to a real-time control application, such as a video capture card, a fieldbus interface, or an Ethernet network interface controller (NIC) designated for communication with real-time I/O devices, should not be multiplexed between virtual machines. This notion of applying I/O exclusively to a specific virtual machine is essential to guarantee real-time responsiveness, because it allows a real-time virtual machine to have direct physical access to its dedicated I/O. Without exclusive physical assignment of pertinent I/O one runs the risk of waiting indeterminately for access to key devices, which can cause failure of time-deterministic processes depending on that I/O operation.

Having consolidated two or more operating systems on one platform, the VMM can facilitate inter-OS communication using shared memory. Since one or more CPUs are dedicated to each operating system, message signalling can be provided by way of inter-processor interrupts (IPIs) between the CPUs. This shared-memory interface is capable of provid-

ing very high performance communication. More complex protocols may then be built on top of this base. Virtual devices can be used as the interface for inter-OS protocols, especially for integration within legacy operating systems and applications. In this case, the inter-OS protocol can be implemented entirely within a virtual hardware interface

For example, all guest operating systems can be configured to share a single area of shared memory to post common data. After each guest updates its data structure it signals the other guests to indicate that a data update has occurred. The shared memory can be presented via a virtual PCI device interface and a register that generates an inter-OS signal via IPI to the other guest operating systems. The VMM, also called a real-time hypervisor, allocates the shared memory area and presents a virtual PCI device to each guest via the guest OS PCI configuration space. Each virtual PCI device includes information on how each guest OS accesses the shared memory and a register to generate the inter-OS signals. In the example above, the virtual PCI device presents two memory ranges to each guest. The first memory range, pointed to by PCI configuration register BAR0, maps the shared memory buffer to an accessible address range within the guest OS.

The second range, pointed to by BAR1, presents an I/O address to the guest OS. When an application within the guest OS accesses the BAR1 I/O address, a trap is made into the virtual device driver hosted by the hypervisor. The virtual device driver then injects a virtual IRQ into the target guest OS which responds by accessing the shared memory area for updated data.

Modern processors contain instructions designed to efficiently perform DSP arithmetic operations. As a group these are known as SIMD instructions (single instruction, multiple data). On Intel architecture processors the SIMD instructions are referred to as the MMX and SSE instructions. MMX instructions were introduced first and are limited to integer operations. SSE instructions were introduced later and can efficiently handle floating point DSP operations and vector arithmetic. These SIMD instructions are ideal for implementing digital filters, the basis of complex digital control algorithms, pattern recognition systems, and video streaming and mixing applications.

Rather than dedicate a costly DSP board inside a dedicated RTOS platform and/or a GPOS box to implement a complex control system, it makes more economic sense to combine these functions into a single hardware platform. Where previously a GPOS system was used for human interface and enterprise network access, an RTOS box for primary machine control, and a DSP board for high-performance data acquisition and filtering, it is now possible to combine them on a single multi-core processor system. By applying asymmetric multiprocessing, one can simultaneously operate three virtual machines on a single platform, dedicating a CPU core to each function, without the cost and complicated development and test associated with multiple separate pieces of hardware. ■

Product News

■ **Lauterbach: integration of TRACE32 and Telelogic Rhapsody**

Lauterbach announced the availability of an integration of their debugging environment, TRACE32 with the UML, SysML, and AUTOSAR based Model-Driven Development environment, Rhapsody from Telelogic. The integration includes a mutual navigation of source code, debugging functionality inside the UML tool, and the animation without an external interface.

News ID 346

Building applications for network-enabled Linux-based devices

By Günter Obiltschnig, Applied Informatics Software Engineering

The POCO C++ libraries are a collection of portable, cross-platform C++ class libraries that can be used to create network-centric applications in technical and embedded computing. The core libraries are developed in an open source project.

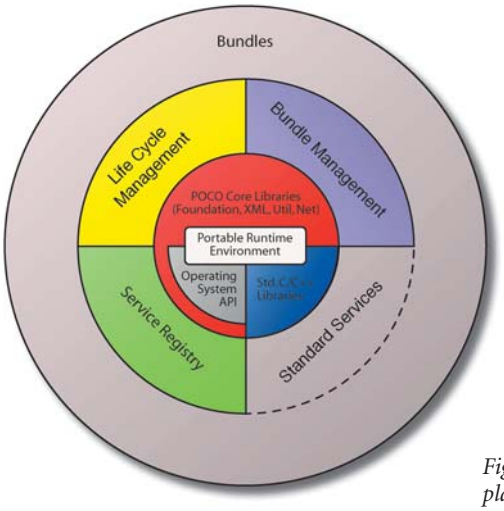


Figure 1. Open service platform layered architecture

Software for network-enabled smart devices becomes ever more sophisticated and complex. This complexity is driven by the many requirements that these devices must satisfy. For example, a built-in web server used for device configuration, administration and monitoring is considered a standard feature today. In the past, a device that needed to exchange data with a host computer typically had a serial interface and used a simple, often home-grown protocol for sending messages over this interface. In modern devices, the role of the serial interface for host communication is being taken over by an Ethernet interface. Instead of a simple binary protocol, a rich XML-based protocol is commonly used. A glimpse into the near future shows us that devices will soon be required to provide and consume web services based on SOAP (Simple Object Access Protocol).

Traditionally it was common that vendors of embedded devices created the entire software stack themselves, except maybe for the operating system, if one was used at all. This is no longer feasible today. Especially if Linux is used as operating system for the device, a rich collection of free open source libraries and applications is available for developers to chose from, covering features like XML parsing, network protocols (HTTP, etc.) and so on. While these libraries can be used without paying licence fees, their use is nevertheless not without cost. Licensing, legal (GPL) and support issues aside, these third-party libraries and applications must first be integrated into the development environment and the build system, and developers must learn to use them efficiently. Especially if third-party C++ libraries are used, different libraries often provide overlapping functionality.

For example, most of the class libraries that have been developed before the C++ standard library became widely available come with their own string and collection/container (arrays, lists, etc.) classes. This is damaging in three ways. First, scarce memory is wasted for the same functionality being unnecessarily implemented multiple times. Second, code that uses these libraries must often convert data structures from one representation to another. Third, developers have to invest additional learning effort, knowing the details and intricacies of all libraries. This gets worse if different libraries use different coding styles and design philosophies.

In 2004, Applied Informatics started with the development of the POCO C++ libraries. The goal was to come up with a collection of portable, cross-platform C++ class libraries that can be used to create network-centric applications in technical and embedded computing. Currently at version 1.3, these libraries cover a wide range of functionality, from basic operating system abstraction (filesystem access, multithreading, etc.), to XML parsing, to networking (TCP and UDP sockets, HTTP server and client, FTP, SMTP, POP3, SSL/TLS, etc.). The core libraries are developed in an open source project (http://pocoproject.org), and are freely available for both open source and commercial use, under a very liberal open source licence. Building on top of these open source libraries, Applied Informatics provides additional libraries and tools. This includes support for Fast Infoset, a very efficient binary encoding of XML, standardised by the ITU and ISO as ITU-T Rec. X.891 and ISO/IEC 24824-1. With Fast Infoset, considerable performance improvements can be gained if XML formatted data is to be sent across a network. From the developer's perspective, it is just the same whether XML data is serialised as plain-text XML, or binary Fast Infoset, as the Fast Infoset library is completely integrated with the XML library. Fast Infoset is a great choice if data is to be exchanged with host applications written in Java or .NET, as Fast Infoset implementations are readily available for these platforms. The interoperability of the different implementations has been verified.

The POCO Netconf library provides support for the NETCONF protocol, a standardised (RFC 4741) protocol for secure remote device

configuration and management. The NETCONF protocol has been developed by the IETF NETCONF working group, with the participation of companies like Cisco, Juniper Networks, Nortel and Motorola. With the NETCONF protocol, XML-based device configuration data can be securely transferred to, or retrieved from a device in the field. Also, different configuration sets can be managed in a standardised way.

POCO remoting is a comprehensive framework for building distributed applications and SOAP-based web services. POCO remoting can turn an existing C++ class into a network or web service. For this, a C++ class declaration has to be annotated with special comments. A code generator uses these annotations to generate the client-side and server-side marshalling, proxy and stub code. Different transport protocols are supported, e.g. a highly efficient binary transport protocol and an XML-based protocol that supports SOAP and WSDL and can be used to communicate with Java and .NET-based host applications. Figure 2 shows a C++ class declaration containing annotations for turning this object into a web service.

The POCO open service platform (OSP) is a C++ based middleware providing a service-oriented and component-based environment for developing, deploying, running and managing modular network-based applications. As such, OSP is to C++ to what the OSGi service platform, developed by the OSGi Foundation (http://www.osgi.org) is to Java. At the core of OSP lies a powerful software component model based on the concept of bundles. A bundle is a deployable entity, consisting of both executable code and the necessary configuration, data and resource files (e.g. HTML pages) required for running the code. Bundles extend the functionality of an application by providing services to other bundles, end-user functionality or web services. A central service registry allows bundles to discover the services provided by other bundles. Bundles can be added, updated, started, stopped or removed from an application without the need to terminate and restart the application. With the optional web-based administration console provided by OSP, this can even be done remotely with a web browser.

This component-based architecture of OSP addresses an increasing problem in software development: the large number of application configurations that need to be developed and maintained. The standardised OSP component architecture simplifies this configuration process significantly. The POCO open service platform is based on a layered architecture, depicted in figure 1. At the core of OSP is the portable runtime environment, consisting of the C and C++ standard libraries and the POCO core libraries (Foundation, XML, Util and Net). Layered above the portable runtime environment is the OSP framework, consisting of service registry, life Cycle management, bundle management and standard services. Application-specific bundles based on the OSP Framework implement the actual application logic.

The portable runtime environment sits at the center of the OSP architecture. Based on the C and C++ Standard Libraries, as well as on the POCO core libraries, it provides platform-independent low-level services to the upper OSP layers, such as: access to the file system, multithreading support, shared library and class loading, notifications and events, logging and error reporting, XML parsing, configuration data handling, TCP/IP sockets and support for various network protocols (HTTP, FTP, SMTP, POP3, etc.), various utility classes and functions.

By isolating applications from the operating system interfaces, the portable runtime environment makes it possible to write applications that can be compiled for and run on different operating system platforms and processor architectures, all from the same source code. This is great for projects consisting of both embedded Linux systems and Windows or Unix-based server or client applications like building au-

```
//@ serialize
struct Measurement
{
    Poco::Timestamp time;
    double          temperature;
    double          humidity;
    double          pressure;
};

//@ remote, synchronized=true
class WeatherStation
{
public:
    void getMeasurements(
        int count,
        std::vector<Measurement>& measurements);

    // ...
};
```

Figure 2. C++ web service built with POCO Remoting

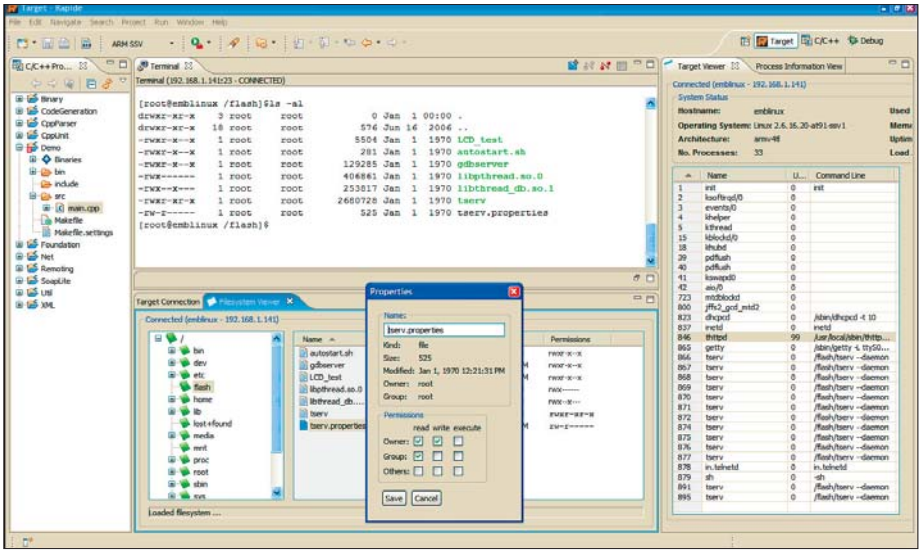


Figure 3. Rapide IDE

tomation or industrial automation systems. The same code base can be used on all platforms, significantly reducing the development effort. The service registry allows a bundle to register its services to make them available to other bundles, as well as to discover the services provided by other bundles. Since bundles can appear and disappear in an application at any time, the service registry also provides notification mechanisms so that a bundle can be informed when another bundles it depends on is removed from the system. This also creates the foundation for building smart devices that are capable of being upgraded with new software features, even when already deployed in the field. OSP comes with a number of standard services implementing commonly required features. Examples include a built-in HTTP/HTTPS server, user authentication and authorization, support for sending email messages, web-based bundle management and remote monitoring, as well as a configuration and preferences storage service.

All POCO libraries are implemented in highly portable ANSI C++. Both the POCO libraries, as well as applications built with POCO can be easily ported to another operating system with minimal effort, usually just by recompiling. The hardware requirements are modest. Popular ARM-9 based microcontrollers (e.g., the Atmel RM9200) with a minimum of 32 MB of RAM are a perfect target platform for deploying POCO-based applications. POCO is being successfully used by customers in different industries, ranging from industrial automation, data acquisition, building management, to medical devices.

Applied Informatics has also developed the Rapide integrated development environment for embedded Linux application development. Rapide is based on the Eclipse platform and the Eclipse C/C++ development tools (CDT). Applied Informatics has added plug-ins that simplify working with embedded Linux targets. For example, development boards connected via TCP/IP are automatically detected by Rapide. It is thus not necessary to know the board's IP address to connect the IDE to it. Finding out a board's IP address can be a tedious task, since a board usually receives its IP address via DHCP, and the board has no way of communicating the IP address it has received to the user. Rapide also integrates a target filesystem browser that supports file transfers to and from the board via drag-and-drop, as well as a target viewer that displays information about the target's operating system status and the processes running on the target.

Remote debugging is fully supported. The executable is automatically transferred to and launched on the target. Rapide includes a GCC-based toolchain for cross-compiling. The POCO C++ libraries are fully integrated, and there are project creation wizards available that automatically create skeleton applications of various kinds. With Rapide, the time from connecting the target board to running the first application with a built-in web server is less than five minutes. Rapide is available for both Windows and Linux hosts. With Rapide running on a Windows host system, no separate Linux system is necessary for embedded Linux application development. This is a huge step forward when compared to existing solutions that require a Linux host system for application development. With POCO and Rapide, a comprehensive and fully integrated platform for the development of network-enabled smart devices based on embedded Linux is available.

This platform can greatly reduce the development risks and costs and improve significantly the productivity of developers. This will lead to a faster time-to-market for new embedded Linux based products. POCO and Rapide are used with great success by customers in different industries, ranging from industrial automation, data acquisition, building management, to medical devices. ■

Powerline communications – the future for smart homes?

By Jon Burroughs, Microchip Technology

Forget WiFi, when it comes to networked communications for a smarter home, look no further than the power socket. This article explores the realities of powerline communications.

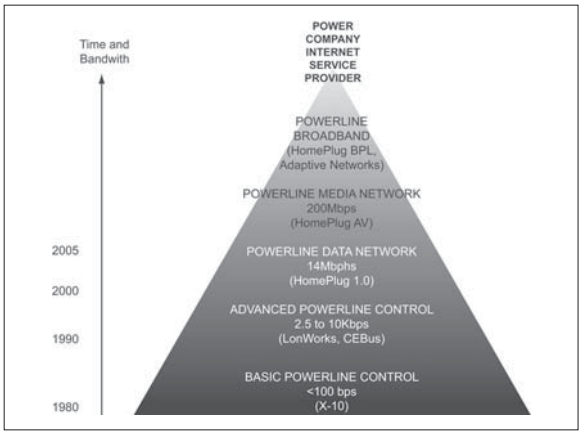


Figure 1. The powerline communications pyramid

Today, they are humble domestic sockets, but tomorrow, the 240V AC power socket could be the powerhouse for sophisticated control and data communication networks, audio/video media streams and even broadband internet access in the emerging Smart Home market.

The potential for powerline communications is enormous. Communications via domestic power sockets could allow data and peripherals to be shared between multiple personal computers; it could allow electricity suppliers to offer broadband connections, or enable a fridge to automatically alert a repair-man when it detects a fault. Powerline communications could even make it possible for energy suppliers to remotely manage loads, effectively giving them to ability to control rolling power-cuts or enable power-hungry appliances such as tumble driers only during off-peak hours, in exchange for lower electricity rates.

Whilst the reality of some of the possible scenarios for powerline communications would not pass the acid test of consumer acceptance, it is wholly possible that, in the near future, powerline communications could be a viable technology for providing the new Smart Home market with networked communications. According to analysts at IMS Research, the percentage of home appliances enabled with some form of networked communication is expected to increase tenfold by 2008. Powerline communications is one of a number of technologies vying for a share of the projected growth in the Smart Home market.

Wireless (Wi-Fi) connectivity, networking over coax and phone lines with the Home Phone Networking Alliance (HomePNA), or even using copper plumbing as a possible carrier for skyscraper condominium local-area networks (LANs) are all contenders for networked communications in the Smart Home. Each of these technologies offers advantages for different applications, but they all share the common goal of providing new networked communication without incurring the expense of new cabling. Whilst the wireless solution promises 'no cables', powerline communications is based on the idea that there will be 'no new cables'.

Potentially, there are four levels of networks which could be carried via the powerline, each of which can be characterised by differences in bandwidth, function and range (see figure 1):

- Powerline control networks, such as those used in home automation, are typically low bandwidth, house-sized networks with low data rates of under 100 bps to 10 kbps.
- Powerline data networks are an alternative to traditional dedicated LANs used to enable data- and peripheral-sharing between multiple personal computers. They support data rates from 1 Mbps through to 15 Mbps.
- Powerline media networks enable multiple video/media streams, and other data, over rates approaching 200 Mbps, with an internet connection via a residential gateway.
- Powerline broadband networks extend communication beyond the home. With sufficient bandwidth, power-line broadband networks could not only enable automated meter reading, but also compete with telephone, cable and satellite providers for internet services.

Of these four categories, control networks are the most realistic implementation of powerline communications. These offer a low marginal cost, demonstrable usefulness and can be implemented with the common, non-proprietary X-10 communication protocol. Although slow by digital standards, X-10 is the most commonly-used technology in home automation, sending low-bandwidth communication signals over powerlines. With a vast installed base, X-10 offers the potential to add low-bandwidth communications with the added confidence that even developers of high-end power-line communications networks are likely to ensure that their technologies are compatible. It is the combination of the X-10 protocol and the ubiquitous 8-bit microcontroller that makes powerline communications a cost-effective option. The X-10 communication protocol works by sending 120 kHz bursts timed with the zero-

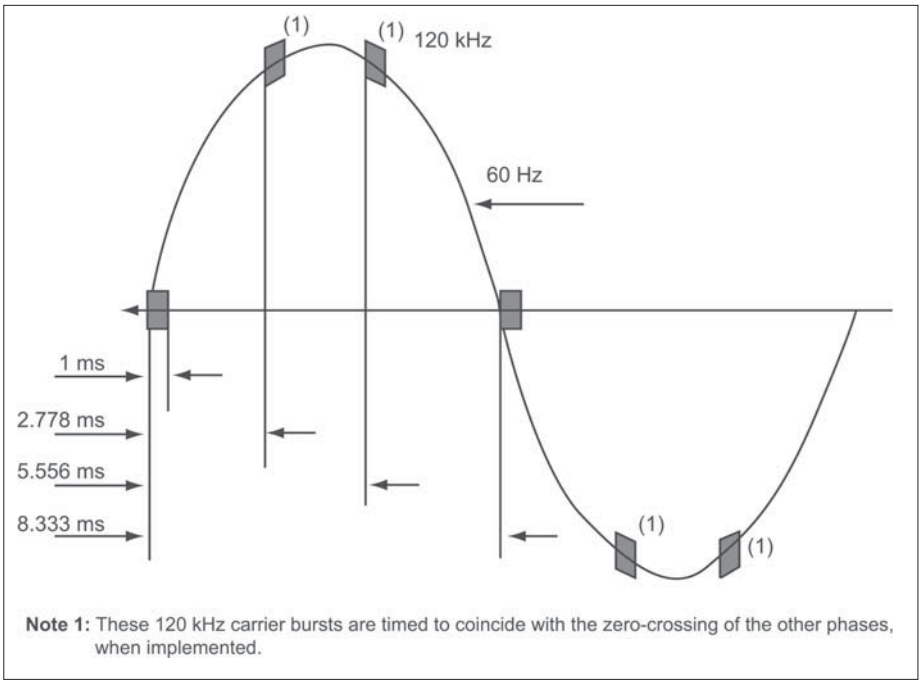


Figure 2. X-10 carrier burst timing

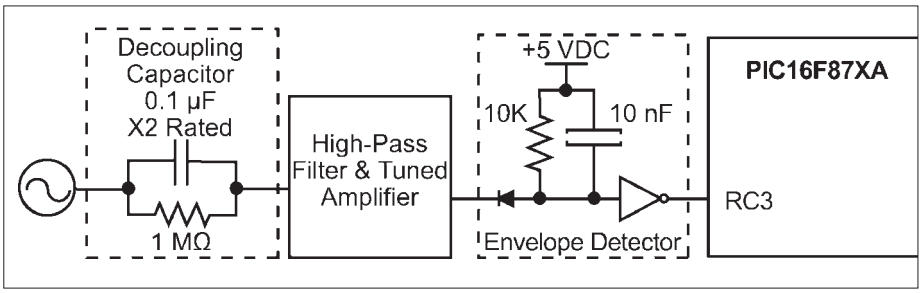


Figure 3. 120 kHz signal detection circuit

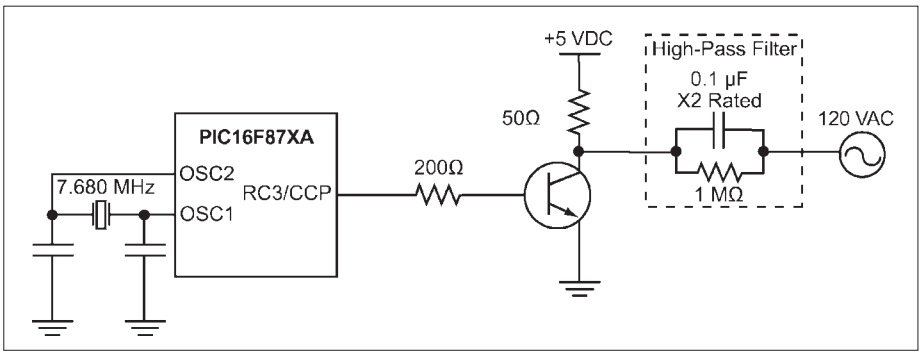


Figure 4. 120 kHz signal generation circuit

crossings in the powerlines to represent digital information (see figure 2). A binary zero is represented by the lack of the 120 kHz burst. In a three-phase distribution system, it is possible for a receiver to be located on a different phase than the transmitter. X-10 therefore specifies that the transmitter sends the 120 kHz burst three times, once at the zero-crossing, and twice more to correspond with the zero-crossings of the other phases. In order for an X-10 signal to be propagated reliably from one phase to another, either a capacitor or active repeater is installed between the phase wires. A complete X-10 message is composed of a start code (1110), followed by a house code, followed by a key code. The key code may be either a unit address or a function code, depending on whether the message is an address or a command. Tables 1 and 2 depict the possible values of the house and key codes. When transmitting house and key codes, two zero-crossings are used to transmit each bit as complementary bit pairs, with zero represented by 0-1, and one represented by 1-0. For example, in order to send the house code A (0110) the code is transmitted as the complementary bit-pairs, 01101001. Since house and key codes are sent using the complementary format, the start code is the only place where the pattern 1110 appears in an X-10 data stream. The 5-bit key code takes 10 bits to represent in the complementary format. As the last bit of the key code is always zero for a unit address, and one for a function code, the last bit of the key code can be treated as a suffix that denotes whether the key code is a unit address or function code.

A complete block of data consists of the start-code, house-code, key-code and suffix. Each data block is sent twice, with 3 powerline cycles, or six zero-crossings, between each pair of data blocks. To turn on an X-10 module assigned to house code A, unit 2, for example, the following data-stream would be sent on the power line, at one bit per zero-crossing:

First, the address is sent twice:
Start Code: 1110
House A: 01101001
Unit 2: 10101001
Suffix: 01
Then there is a pause for three cycles or six zero-crossings:
000000
The command is then sent twice:
Start code: 1110
House A: 01101001
ON: 01011001
Suffix: 10
There is another pause for three cycles (six zero-crossings) before the next block of data is sent. Three extended codes support data and control, security messages and meter reading. Alternatively, the protocol can be revised to provide better security and more address capability. One house code could, for example, be re-assigned to indicate that the extended protocol would follow, and after that sequence the new message format could be used.

Effectively, powerline communications (PLC) turns every power outlet into a potential communications socket and yet the hardware required to implement an X-10 system comprises just a zero-crossing detector, a 120 kHz carrier detector and a 120 kHz signal generator. The microcontroller resources required to implement these functions are available on virtually all Microchip Technology's 8-bit PIC microcontrollers. The following example is based on a PIC16F877A 8-bit microcontroller which has an external interrupt on RBO for detecting the zero-crossing; a CCP1/Timer 2 in PWM mode for 120 kHz modulation; a digital input pin for the carrier-envelope detect circuit; a Timer0 interrupt for the 120 kHz envelope timing; and minimal memory of under 1 Kbyte of programme memory for the X-10 firmware

functions and less than 100 bytes of data memory for the X-10 firmware variables.

The zero-crossing detector is implemented using the external interrupt on the RB0 pin and just one external resistor to limit the current into the PIC microcontroller. In North America the Vrms is 117VAC, and the peak line voltage is 165V. Selecting a resistor of 5 Mohms creates a current which is well within the clamp-current capacity of the PIC microcontroller I/O pin:

Ipeak= 165V/5 Mohms = 33 µA

Input protection diodes, integrated into the PIC microcontroller's I/O pins, clamp any voltage higher than Vdd or lower than Vss. Therefore, when the AC voltage is in the negative half of its cycle, the RB0 pin is clamped to Vss-0.6V which is interpreted as a logic zero. When the AC voltage rises above the input threshold, the logical value becomes 1. In this example, RB0 is configured for external interrupts, and the input buffer is a Schmitt trigger. This makes the input threshold 4V on a rising edge (0.8 Vdd) and 1V on a falling edge (0.2 Vdd). On each interrupt, the interrupt edge select bit within the OPTION_REG register is toggled, so that an interrupt occurs on every zero-crossing. The following equation calculates when the pin state changes relative to the zero-crossing:

V= Vpk * sin(2*pi*f*t)

With a 165V peak voltage and 60 Hz frequency, RB0 goes high at around 64 µs after the zero-crossing on a rising edge, and goes low around 16 µs before the zero-crossing on a falling edge.

The carrier detection circuit for the 120 kHz signal on the AC powerline comprises a decoupling capacitor, a high-pass filter, a tuned amplifier, and an envelope detector (see figure 3). The impedance of the capacitor is:

$Zc = 1/(2*\pi*f*C)$

Therefore, a 0.1 µF capacitor presents a low, 13Ω impedance to the 120 kHz carrier frequency, but a high, 26.5 KΩ impedance to the 60 Hz powerline frequency. This high-pass filter allows the 120 kHz signal to be safely coupled to the 60 Hz power line, and doubles as the coupling stage of the 120 kHz carrier generator. Since the 120 kHz carrier frequency is much higher than the 60 Hz powerline frequency a straightforward RC filter can be used to pass the 120 kHz signal and completely attenuate the 60 Hz. For a simple high-pass filter, the –3 db breakpoint is:

$f3\ db = 1/(2*\pi*R*C)$

Therefore, with a capacitance is 100 pF and resistance of 1 MΩ:

$f3\ db = 1/(2*\pi*150\ pF\ *33\ K\Omega) = 32\ kHz$

This f3 db point assures that the 60 Hz signal is completely attenuated, while the 120 kHz signal is passed through to the amplifier stages. Next, the 120 kHz signal is amplified using a series of inverters configured as high-gain amplifiers. The first two stages are tuned amplifiers with peak response at 120 kHz. The next two stages provide additional amplification. The amplified 120 kHz signal is passed through an envelope detector formed with a diode, capacitor and resistor. The envelope detector output is buffered through an inverter and presented to an input pin (RC3) on the PIC16F877A microcontroller. On each zero-crossing interrupt, RC3 is checked within the 1ms transmission envelope to see whether or not the carrier is present.

The presence or absence of the carrier represents the stream of ones and zeros that form the X-10 message. The 120 kHz carrier signal is generated either with an external oscillator circuit or by using one of the PIC microcontroller's Capture/Compare/PWM (CCP) modules. The CCP1 module is used in PWM mode to produce a 120 kHz square-wave with a duty-cycle of 50%. A 7.680 MHz oscillator is used in this example for the CCP to generate precisely 120 kHz. After initialisation, the CCP1 is continuously enabled, and the TRISC bit for the pin is used to gate the PWM output. When the TRISC bit is set, the pin is an input and the 120 kHz signal is not presented to the pin. When the TRISC bit is clear, the pin becomes an output and the 120 kHz signal is coupled to the AC power line through a transistor amplifier and capacitor (see figure 4). The signal generator uses the high-pass filter in the detection circuit shown in figure 3.

To be compatible with other X-10 receivers, the maximum delay from the zero-crossing to the beginning of the X-10 envelope should be in the region of 300 µs. Since the zero-crossing detector has a maximum delay of approximately 64 µs, the firmware must take less than 236 µs after detection of the zero-crossing to begin transmission of the 120 kHz envelope. The number of potential applications for powerline communications in Smart Homes is vast.

Smart coffee pots could, for example, be set to turn on minutes before the alarm-clock; smart air conditioning could use individual room thermostats to control the airflow into each room via variable-speed fans located in the ventilation ducts, and a smart clothes iron could automatically be turned off when the lights are turned off, when the home alarm system is armed, or when it is left vertical for more than one hour. All these Smart Home appliances, and more, can be realised within the existing X-10 installed base with the relatively simple addition of an 8-bit microcontroller and a few external components. ■

A managed memory subsystem for embedded systems

By Vijay K. Devadiga, Silicon Storage Technology

The All-in-OneMemory decribed in this article provides an easy-to-use, completely managed memory subsystem for code and data storage in a single package.

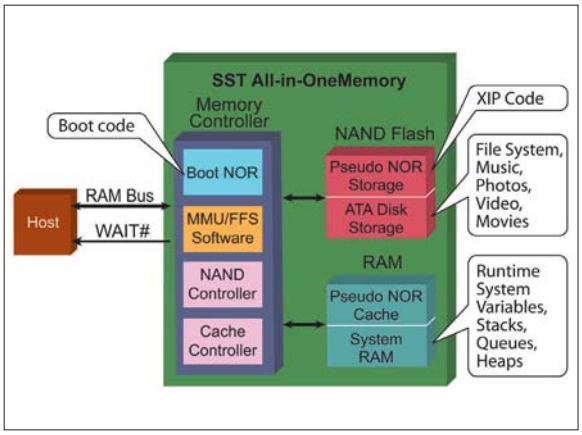


Figure 1. Block diagram of the various functions of All-in-OneMemory

The increasing multimedia functions in cellphones and other digital consumer devices have caused system designers to re-evaluate the memory subsystem in these devices. The requirements of increasing functionality, shrinking physical size and reducing overall system cost in consumer electronics devices can create conflict in a system that continues to use a conventional code and data storage solution. The All-in-OneMemory solution from Silicon Storage Technology (SST) resolves these conflicts by providing system designers with an easy-to-use, completely managed memory subsystem for code and data storage in a single package. This solution has taken the complexity out of hybrid memory architectures by enabling a single-bus memory subsystem that uses an industry standard RAM (PSRAM/SDR/DDR) host interface to provide seamless integration to the host processor. By re-architecting the conventional memory subsystem, All-in-OneMemory addresses the cost and time-to-market needs of system designers working on mobile and consumer electronics devices.

It is a highly integrated, completely managed memory subsystem that includes code storage, data storage, and system RAM functions, which are managed by an intelligent built-in memory controller, in a single package and on a single bus. This solution blends the key benefits of NOR (fast read, random access), NAND (lower cost, higher density), and RAM (simple bus operation) into a unified architecture to enable multiple gigabytes of execute-in-place (XIP) code storage and a memory-mapped ATA NAND disk (mATA) area satisfying the growing storage needs of embedded applications.

The SST solution consists of a memory controller, NAND flash and random-access memory (RAM) in a single device. See figure 1 for the block diagram. The memory controller features embedded SuperFlash NOR blocks for boot code, a flash file system (FFS) for NAND flash management, NAND controller with hardware ECC, and a cache controller for pseudo NOR (PNOR) that emulates high-density NOR flash. The NAND controller incorporates robust hardware error detection and correction circuitry (ECC) for both MLC and SLC NAND. The memory solution has four memory areas (see figure 2) each selectable by a separate chip enable input. The memory areas are: NOR, Pseudo NOR (PNOR), RAM, and mATA. Except for the NOR area, the size of these memory areas is configurable by the host software. The NAND flash is used as the non-volatile storage media for both the PNOR and mATA areas. The RAM is used for PNOR cache and system RAM. The technology also offers a three-chip enable configuration for systems that cannot utilise the default four-chip enable configuration.

The NOR area is directly mapped to the instant-on, embedded SuperFlash NOR block in the memory controller and offers fast boot time. The NOR area is available for host access immediately after power-on and is suitable for storing boot code and any time-critical code and data. The NOR area also offers many advanced data protection features for secure boot and for protecting code and data content. The data protection features in the NOR area include: OTP secure blocks, volatile block protection, non-volatile block protection, password protection, hardware boot block protection, and 512-byte security-ID with an OTP lock. These advanced data protection features enable a wide range of secure applications such as secure transactions and digital content subscription.

The expandable non-volatile PNOR area emulates high-density NOR by using NAND for non-volatile storage and RAM as cache memory (see figure 3). The NAND flash offers only sequential access, while NOR flash offers random execute-in-place (XIP) access. Having RAM cache in front of NAND flash converts sequential access to random access, thereby emulating the NOR read function. The PNOR area, which uses NAND as non-volatile storage media, is a cost-effective solution that offers multiple gigabytes of linear addressable XIP code storage suitable for storing operating

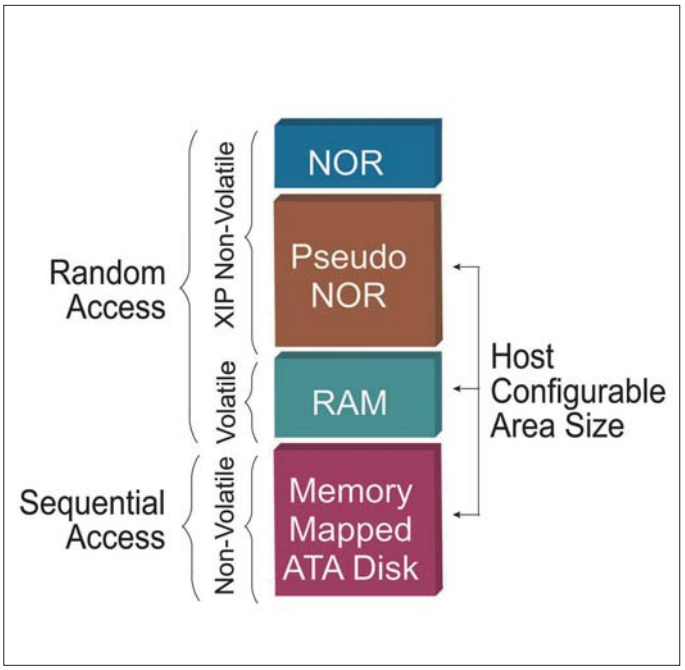


Figure 2. All-in-OneMemory host memory map

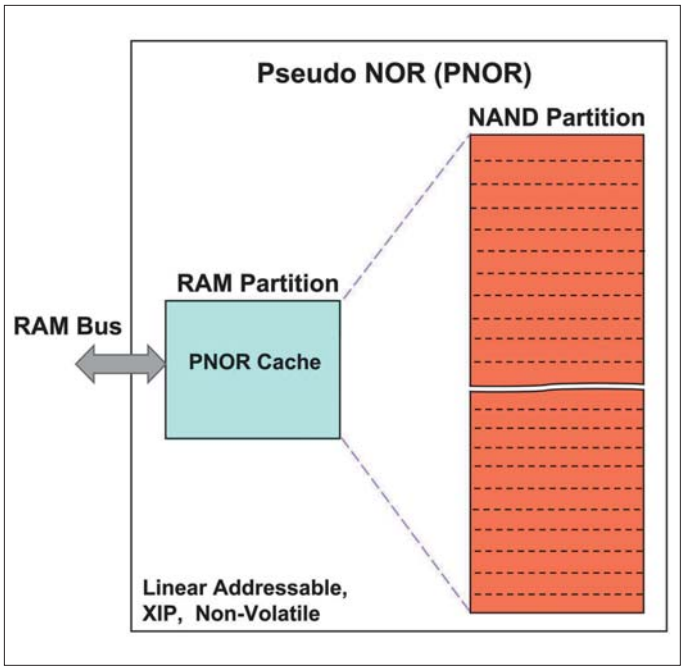


Figure 3. PNOR block diagram of the memory solution

system, protocol stack, and other application code. The PNOR can replace high-density NOR flash in a conventional memory subsystem and enables designers to take full advantage of the rapidly dropping NAND price and the ever-increasing densities that NAND flash offers. The PNOR area uses a simple RAM type of read/write operation that simplifies both hardware and software design on the host side. For better read/write performance, the PNOR area supports both page and burst mode.

The size of the PNOR area and the size of the PNOR cache are configurable by the host software. Because the performance of PNOR area depends on the size of PNOR cache, host software can increase the PNOR cache size, up to the size of the built-in RAM, to improve application performance. The device has many cache performance monitoring parameters and supports different cache replacement algorithms. The host software can read these cache monitoring parameters to determine the cache hit/miss ratio. Based on the ratio, the host software may reconfigure the PNOR cache size and/or may select a suitable cache replacement algorithm to optimise the cache performance.

The cache controller handles on-demand paging and supports two different paging modes: static paging mode and dynamic paging mode. The cache coherence operations are totally managed by the All-in-OneMemory controller without any host intervention. In the event of low battery power indication, system power-down, etc, the host software may issue flush cache commands as required. The built-in RAM is divided into two partitions: PNOR cache and system RAM. The size of these partitions is configurable by the host software. The first partition is used as cache for the PNOR area and the second partition is available for the host as system RAM for storing run-time variables, stacks, queues, heaps, etc. The RAM area is directly mapped to the system RAM partition in the RAM and offers direct access from the host. The RAM area also supports page mode and burst mode for better read/write performance.

The mATA area offers ATA like disk storage and uses the standard ATA protocols. This memory area can address up to 128 Gbytes of data storage and supports only block transfers with the minimum block size of one sector (512-byte) as in standard ATA protocol and is suitable for storing music, video, movie, photo, and data files. In All-in-OneMemory, the task file registers are memory mapped, not I/O mapped as in the standard ATA device, hence it is named as memory-mapped ATA (mATA). By memory-mapping the task file registers, it is able to offer ATA like disk storage on the standard RAM bus without the need for the standard ATA interface.

The device is highly configurable with many host configurable parameters and registers. The host configurable parameters (PNOR area size, PNOR cache size, etc) are stored in the non-volatile memory registers and the default value of these parameters is configured during manufacturing at the SST factory. The host software can reconfigure a parameter by issuing vendor-specific ATA commands. Moreover, the same device can be targeted for different applications by just reconfiguring the host configurable parameters and this can help to significantly reduce the development time and cost, and also speed time-to-market. The device also has many bus configuration registers that define how it interfaces with the host bus. The host can program the bus configuration registers to select the asynchronous or synchronous mode of operation, access latency, WAIT output polarity, WAIT configuration, burst length, etc.

The device is highly scalable and offers the same interface and ball footprint for different densities. This makes moving from a lower density device to a higher density device (and vice versa) easier on the same footprint without the need for any hardware change. The device is highly integrated and is fully tested and qualified. This helps to reduce the test time and test cost at the system level. ■

Product News

■ **Digi-Key: broad expansion of Infineon product line**

Electronic component distributor Digi-Key announced the broad expansion of its Infineon Technologies product offering to include Infineon's power semiconductors, discretes, sensors, wireless control products, communication ICs and microcontrollers. Within the Infineon microcontroller (MCU) family, Digi-Key now offers the XC166 and C166 16-bit, XC800 and C500 8-bit, and TriCore 32-bit families of MCUs. The XC166 family with pipelined 16-bit core architecture, optimized with real-time embedded control peripherals, offers a wide range of performance variations.

News ID 295

Development tools are key to FPGA deployment in SoC designs

By Mike Thompson, Actel

A full suite of proprietary and third-party high-level tools allows designers to combine the ARM Cortex-M1 with secure flash-based Actel FPGAs to implement complex SoC designs.

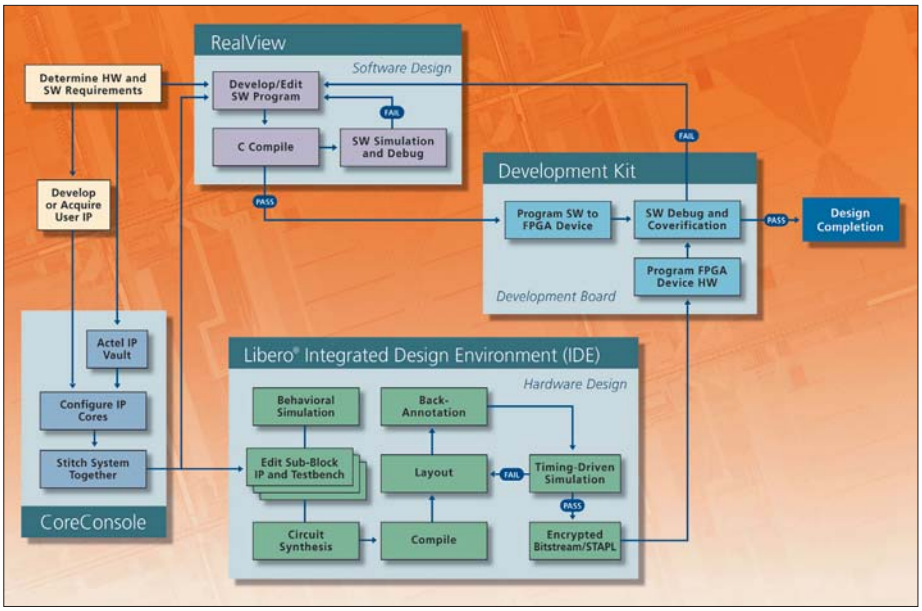
According to CMP's '2006 State of the Embedded Market Survey', the FPGA market will be impacted by several important trends regarding processor usage, IP reuse and the overall use of programmable logic in embedded projects. While the report states that approximately one-fifth of todays FPGA designs currently use a soft processor, it also asserts that the majority of FPGA designers see themselves using soft processors, such as those offered by ARM, in the future. The study also suggests that approximately 75 percent of all embedded designs feature some degree of intellectual property (IP) reuse. Finally, the study reports that 81 percent of embedded projects now feature an FPGA, underscoring the continued move towards FPGAs and away from ASICs.

While these trends are undoubtedly good news for the FPGA market, there is little doubt that there are also associated challenges. For example, typical designs routinely reach size and complexities previously only attainable in more traditional ASIC-type SoC devices. As a result, they require the use of high-level software tools to allow designers work at an intuitive higher level to implement the design in an FPGA. As IP reuse becomes more popular, the same tools need to offer the ability to quickly and simply integrate both in-house and third-party IP offerings. Also, designs need to be simulated and debugged, and software written to complete the application – often at the same time as the hardware design is progressing. The use of IP also raises the bar for the devices themselves. Valuable soft IP needs to be deployed securely, with protection from tampering and theft during development, manufacturing, and in the field. This is one of the reasons why providers of high-value IP, such as ARM, have traditionally been wary of supplying their products in soft form for use in FPGAs. All of this illustrates the significance of Actels recent launch of the ARM Cortex-M1 soft processor and its accompanying suite of tools, including the CoreConsole IP deployment platform (IDP).

As the leading 32-bit processor, the availability of an ARM processor designed specifically for FPGA implementation is more or less a prerequisite in the continuing displacement of ASICs in SoC applications by FPGAs. Actel's non-volatile, flash-based ProASIC3 FPGAs and Fusion programmable system chips (PSCs) allow commercial IP to be deployed securely. They are inherently difficult to reverse engineer and do not require an external configuration device from which design data can be read. They also feature an on-chip AES encryption engine with a 128-bit key that allows the soft ARM Cortex-M1 core to be distributed securely and implemented only on specific devices for which its use is authorised. But no less important than the Cortex-M1 and the FPGA device in which it is implemented is the suite of tools that enables their use. Actels CoreConsole IDP works at a level of abstraction above the RTL, allowing designers to perform design entry via a graphical interface. It produces RTL code that is seamlessly exported to the Libero integrated design environment (IDE) for simulation and synthesis as well as the software drivers required for ARM software program development tools.

CoreConsole is a bus-centric front-end design tool that supports rapid SOC assembly and auto-stitching. This enables IP blocks, originated by the user or by third parties, to be automatically connected to a chosen interconnect bus. An IP vault is also included, providing access to Cortex-M1 and other IP cores, as well as third-party IP from Actel CompanionCore partners.

CoreConsole is designed to work at a level of abstraction above the specific interconnect bus standard, processor, subsystem and IP blocks being used. As a result, maximum flexibility is allotted for current and future choices regarding IP. And, by automating the tedious and time-consuming manual 'IP stitching' process, the tool allows designers to focus on the system rather than the components themselves. This frees the designer to conceptually view the system at a higher level, allowing earlier system-level evaluation and reducing overall development time.



CoreConsole and Cortex M1 development tools

This approach reflects the industry's long-anticipated move towards system-level design. The CoreConsole IDP uses an intuitive graphical user interface to give designers system-level control. It also uses methods defined by the tool-flows within the SPIRIT Consortium initiative (structure for packaging, integrating and re-using IP) to ensure industry-standard system-level IP deployment, via underlying structures based on XML code. After the IP is stitched to the bus, the system-level approach is maintained. The design tool allows the designer to get an early view of system functioning. A system interconnect test bench is generated that is exported along with the design for use in Libero IDE. This can then be used to validate and debug the connection of the design within the FPGA fabric. The design tool is tightly integrated with the remaining elements for the Cortex-M1 tool flow, evidence again of a system-level approach. Files generated by the tool can be used seamlessly within the Libero IDE, which provides simulation, synthesis and place-and-route capabilities from Actel and its third-party partners, Magma Design Automation, Mentor Graphics, Synplicity, and SynaptiCAD. Libero gives designers the ability to use the commercial tools with which they are familiar in a streamlined design flow with automated management of all source, design, run and log files. This structure ensures interoperability, permitting seamless passing of all design data between tools from schematic/HDL entry through synthesis, simulation, place-and-route, and device programming. Just as important as an integrated hardware design flow is the ability to start software development as early as possible in the design process, and to pass the necessary information between hardware and software design teams. Therefore the SoftConsole program development environment has been developed with support for Cortex-M1 and the other processor offered for use in Actel FPGAs that can use output from CoreConsole. The tool is based on the widely-used open-source Eclipse IDE and GNU compiler and debugger. It can import memory map and drivers from CoreConsole to ease development and debugging.

The Cortex-M1 processor is also supported by a large and growing number of tools from third-party companies. While both ARM and Actel employ teams of engineers who develop tools to support their products, both companies also engage with many third-party companies that have extensive expertise in developing specific tools. Through partnerships with these third parties, the tools that they offer are optimised for the ARM architecture and use with Actel FPGAs. This increases the range of available tools and gives users a choice as to which tools to use for the development of their application. In addition to the tools from the suppliers, the Cortex-M1 processor is supported by tools from Aldec, CriticalBlue, CodeSourcery, IAR, ImpulseC, Mentor Graphics, Synplicity, and others. Additional tools will be announced as they become available, broadening the range of tools for development of embedded applications. ■

Product News

■ Telelogic enhances model-driven SOA development

Telelogic announces Tau 3.1, a new version of the company's model-driven development solution for the design and development of service architectures and Web services for enterprise applications including those in IT, telecom, defense, finance and transportation. Developers benefit from an architecture-focused approach, which includes executable models that simulate application functionality and behavior from the very beginning of the lifecycle.

News ID 260

■ CMX: four Flash file systems for most processor families

CMX now offers four flash file systems to support a wide variety of embedded designs using 8-bit, 16-bit, and 32-bit processors and DSPs. The following CMX Flash File Systems can be used standalone and can also be integrated with CMX RTOSes, TCP/IP and USB stacks: CMX-FFS is the standard edition flash file system and includes 100% failsafe, multiple volumes, Wear-leveling, directories, boot sector support, and RAM drive.

News ID 290

■ Green Hills: layer 3 IP routing platform for secure networking

Green Hills Software and NextHop Technologies have announced the availability of Green Hills Softwares Platform for Secure Networking (PSN) support of NextHop Technologies GateD advanced layer 3 IPv4 and IPv6 routing software. Based on the Integrity real-time operating system and its IPv4/v6 networking stack, the Green Hills Platform for Secure Networking delivers industry advancements in secure communications and access that allow networking equipment developers to completely secure their devices and the information flowing through them.

News ID 298

■ Freescale: development platform for Power Architecture

Freescale Semiconductor introduces an all-in-one development platform for industrial markets featuring Power Architecture technology and delivered in the COM Express form factor. The solutions MPC8360E PowerQUICC II Pro processor cost-effectively meets or exceeds the performance of the less flexible ASICs and more expensive FPGAs traditionally used to deliver the highly deterministic performance required for industrial applications.

News ID 347

Hitex: tool support for Atmel AT91SAM9263

Hitex's Tantino JTAG debug tool has been extended to support the whole on-chip debug functionality of Atmel's AT91SAM9263 microcontroller. The USB-powered Tantino is operated by the universal HiTOP user interface and provides flash programming and functions as unlimited flash breakpoints, condition-sensitive breakpoints, exception assistant and update on running.

News ID 261

NI: test management software with sequence editor

National Instruments announces TestStand 4.0, the latest version of the company's test management software with streamlined sequence editor that reduces development time and increases ease of use. With the preconfigured step templates in the new sequence editor, engineers can easily reuse preconfigured test templates to quickly build test sequences and save time in test system development.

News ID 275

Green Hills: software development for automotive microcontrollers

Green Hills has announced the availability of its comprehensive ECU software development solution targeting Freescales MPC5510 microcontroller family, built on Power Architecture technology. Components of the Green Hills Software development toolkit include: royalty-free RTOSes, optimizing C/C++ compilers for Power Architecture, MULTI integrated development environment (IDE) with multicore debugger, virtual prototyping platform, TimeMachine tool suite and high-speed Nexus hardware probes.

News ID 316

Renesas: R32C microcontrollers with up to 48 MHz

Renesas Technology announced the R32C/118 group of microcontrollers with on-chip flash memory, as part of the R32C/100 series. This microcontroller group currently consists of two models, and offers over twice the processing performance of the existing M32C/80 series. The group incorporates the R32C/100 CISC CPU core, which is the most powerful CPU core in the M16C Platform and is upward compatible with the existing CPU cores in the set.

News ID 310

Vector: secure flash programming in the vehicle

To assure that flash programming of ECUs is protected against tampering, Vector is integrating cryptographic algorithms from cv cryptovision in its bootloaders. Vector is relying on the use of this tested technology of cryptographic algorithms, which supports security mechanisms required by car manufacturers for software updates.

News ID 332

Wind River Linux selected by automotive consortium

Wind River announced that the Vehicle Infrastructure Integration Consortium (VIIC), has selected Wind River General Purpose Platform, Linux Edition as the platform for developing the Vehicle On-Board Equipment (OBE) for their proof-of-concept activities.

News ID 277

Express Logic: RTOS support for ARM Cortex-M3 MCUs

Express Logic announced that its ThreadX RTOS and NetX TCP/IP networking stack now support Luminary Micros new Ethernet-enabled (LM3S6000 series) Stellaris family of ARM Cortex-M3 microcontrollers. ThreadX also supports the new CAN-enabled (LM3S2000 series) Stellaris family of ARM Cortex-M3 microcontrollers.

News ID 327

Innovasic guarantees supply of FIDO MCUs until 2020

Innovasic Semiconductor announces that it is guaranteeing production of its fido microcontroller family until at least 2020. A signed certificate guaranteeing a fido supply will be included in every evaluation development kit.

News ID 248

Cyan: USB core offers USB 2.0 functionality without external PHY

Cyan Technology launches a functional USB core to its family of low-power, feature rich microcontrollers, the eCOG1X range. With minimal external components the USB device can act as a USB2.0 full-speed (12 Mbit/s) device in host, peripheral or On-the-Go (OTG) modes.

News ID 331

Renesas: microcontroller for high-performance car information systems

Renesas Technology Europe announced the SuperH family SH7775, a SoC solution designed for high-performance car information systems such as next-generation car navigation systems. It incorporates the SuperH family's top-end SH -4A CPU core operating at 600 MHz and is capable of high processing performance of 1 GIPS (giga floating-point operations per second) or more. It is a product that integrates many comprehensive peripheral functions required by car navigation systems.

News ID 303

CMX: TCP/IP and RTOS for Stellaris microcontroller

CMX Systems announced the availability of two RTOSes and a TCP/IP stack for Luminary Micros Stellaris LM3S6000 series of processors. CMX-MicroNet is a TCP/IP stack specially crafted to work with virtually all processors. With a very small ROM requirement ranging from 5kytes to 24kbytes (depending on configuration and processor) and minimal RAM requirements of about 500 bytes plus buffers for packets, the base CMX-MicroNet software package currently includes UDP, TCP, IP, Modem, SLIP, ICMP Echo, IGMP, and Virtual File.

News ID 336

TI: location detection chip for ZigBee wireless sensor networking

Texas Instruments introduced a SoC solution with a hardware location engine targeting low-power ZigBee/IEEE 802.15.4 wireless sensor networking applications. The CC2431, from the companys Chipcon product line, targets applications such as asset and equipment tracking, inventory control, patient monitoring, remote controls, security and commissioning networks. The device is supported by TIs ZigBee protocol stack, Z-Stack.

News ID 348

Microchip: 2MHz, 500 mA switching regulator

Microchip announces the MCP1603 - a 2 MHz, 500 mA switching regulator which provides adjustable and fixed output voltages and is available in Thin SOT-23 and 2 mm x 3 mm DFN packages. The new device is ideal for extending battery life and reducing heat dissipation in a variety of portable, handheld electronic devices.

News ID 271

CML: narrowband wireless IC for M2M applications

CMLs CMX990 is an advanced single-chip RF transceiver with integrated GMSK data modem, which is targeted at global narrowband M2M (machine to machine) products. The half-duplex IC offers a versatile baseband modem, IF and RF processing functions, programmable synthesisers, signal-level setting functions and comprehensive on-chip and peripheral control, monitoring and interfacing. It has a data range of 4 to 16 Kbit/s with selectable BT values at RF frequencies of 400MHz to 1GHz.

News ID 333

Digi: IP gateways for various wireless technologies

Digi International introduced the ConnectPort X product family, a line of IP gateways that provide seamless connectivity of ZigBee, Wi-Fi, cellular and Ethernet traffic to centralised applications and databases. The ConnectPort X8, available now, also features GPS, local storage and easy programmability. The family is augmented by a new family of XBee adapters and routers that provide device connectivity and network integrity.

News ID 357

TI: floating point digital signal controllers

Texas Instruments announced the first floating point digital signal controllers (DSCs). The TMS320F2833x devices provide 300 million floating point instructions per second (MFLOPS) performance at 150 MHz while providing the lower costs typically associated with fixed point processors. Simplified software development common to floating-point processors and the performance boost allows solar power inverters to more efficiently convert energy from photovoltaic (PV) panels, offers better power efficiency and performance to variable speed alternating current (AC) drives and provides greater performance for automotive radar applications.

News ID 322

Elektrobit: acquisition of DECOMSYS strengthens automotive business

Elektrobit Corporation (EB) has entered into a share purchase agreement under the terms of which it shall purchase 100 per cent of the shares in DECOMSYS Beteiligungs GmbH (DECOMSYS). DECOMSYS is a solutions provider for FlexRay, the high performance network communications protocol standard for automotive electronics, and AUTOSAR (Automotive Open System Architecture) next generation standard software.

News ID 360

Tensilica: optimized MP3 decoder

Tensilica announces an optimized MP3 decoder for its Xtensa HiFi 2 Audio Engine and Diamond Standard 330HiFi processor core. This MP3 decoder now runs at lowest power and requires just 5.7 MHz when running at 128Kbps, 44.1 KHz and dissipating 0.45 mW in TSMC's 65nm LP process. This makes the Xtensa HiFi 2 Audio Engine ideal for adding MP3 playback to cellular phones, where current carrier requirements are for 100 hours of playback time on a battery charge, and increasing to 200 hours in the near future.

News ID 249

Linear Technology: three-channel constant current LED driver

Linear Technology Corporation announces the LT3496, a 2MHz DC/DC converter designed to operate as a three-channel constant current LED driver. Each of the LT3496's three channels can drive up to eight 500mA LEDs in series, enabling it to drive up to 24 x 500mA LEDs at efficiencies up to 96%. All three channels are operated by an independent True Color PWM signal, enabling each to be dimmed independently to ratios as high as 3,000:1.

News ID 285

austriamicrosystems: 12-bit magnetic rotary encoder IC

Austriamicrosystems introduces the AS5046, a 12-bit resolution magnetic rotary encoder offering rotation, tilt and vertical distance detection of a magnet placed above the IC. The device can detect vertical distances over several millimeters, offering an solution for contactless, multi-axis human interface devices, such as navigational knobs.

News ID 334

Green Hills: Integrity Workstation support for Intel virtualization technology

Green Hills has announced the availability of enhancements to Integrity Workstation to support Intel Virtualization Technology-enabled processors and chipsets. The Workstation is based on the safety-certified Integrity operating system, and uses Green Hills Padded Cell virtual machine technology to provide an operating environment that supports high criticality applications alongside traditional, unmodified desktop and server operating systems, such as Windows, Linux, and Solaris. Software solutions that incorporate Intel Virtualization Technology enable multiple operating systems to be run simultaneously in their own protected environments.

News ID 339

ECE

Editor
Jürgen Hübner
phone +49(0)8092-2477413
fax +49(0)8092-2477429
jh@iccmedia.com

Contributing Editor
Tony Devereux
devrex@teyboyz.freeserve.co.uk

For Reader Inquiries and Address Changes please contact:
info@iccmedia.com

Sales & Marketing Director
Manfred Blumoser
phone +49(0)8092-2477411
fax +49(0)8092-2477429
mb@iccmedia.com

Claudia Mellein
cm@iccmedia.com

Christiane Lockner
cl@iccmedia.com

Sales Office - UK and USA, Benelux, Scandinavia
Blue Sky Communications
Malcolm Cameron
Kent Innovation Centre
Thanet Reach Business Park
Millennium Way
Broadstairs Kent CT10 2QQ UK
phone +44 (0)1843 609357
fax +44 (0)1843 609358
mc@blue-sky-communications.com

Sales Office - Asia
Jean Cheng, *jean@i-media.com.tw*
Vivian Hung, *vivian@i-media.com.tw*
Innovative Media Information & Services
7F-3, No. 26, Sec. 2, Ming-Quan East Rd.
Taipei 104 Taiwan
phone +886 2 2563 1186

Head Office
ICC Media
ICC Media GmbH
Rauwagnerstr. 5
85560 Ebersberg / Germany

Editorial Office UK
36a Blackacre Road
Theydon Bois
Essex, CM16 7LU

Maxim: OVP controllers drive an external pFET and draw 13 µA

Maxim Integrated Products introduces the MAX4923-MAX4926 family of overvoltage-protection (OVP) controllers. These devices drive an external pFET to provide protection to low-voltage systems against high-voltage faults up to 28V, while drawing a low 13 µA (typ.) supply current. The use of an external pFET enables these OVP controllers to achieve this low power consumption by eliminating the need for an integrated charge pump, which is the main source of current drain in competitive solutions.

News ID 372

Framos: Sony Megapixel CCD for security applications

Framos presents ICX445AKA, Sony's first Megapixel Progressive Scan CCD image sensor for surveillance cameras. The diagonal 6mm image sensor is delivered in a 24-pin DIP plastic package and achieves a resolution of 1.2M effective pixels (1296 × 960). Through the adoption of EXview HAD CCD technology, the new sensor features an exceptionally high sensitivity of 460mV (F5.6) and can be used for day and night shots as well as in connection with so-called 'near infrared' light sources.

News ID 245

Rutronik: Bosch CAN controller for automotive electronics

Rutronik, Distributor for Bosch intergrated circuits and sensors in Europe, started to market the CAN Controller CC770 in the ROHS compliant LQFP44 package. This part is designed for the automotive temperature range (-40 C to +125 C) and offers function and pin out compatibility with the obsolete widely spread Intel 82527. The CC770 serial communications controller performs serial communication according to the CAN protocol.

News ID 288

Densitron enters into partnership with LoGi Engineering

Densitron Display Solutions is pleased to announce its exclusive arrangement with LoGi Engineering, a plastics engineering company with facilities in Malaysia and Singapore, to enhance its complete solutions product offerings. LoGi was established to support the healthcare diagnostics industries worldwide.

News ID 338

QuickLogic: boot from managed NAND solution

The 'boot from managed NAND' solution is a combination of QuickLogic's SDIO host controller and additional intellectual property that performs the boot sequencing function. Managed NAND is an emerging trend in Flash storage technology that integrates both NAND flash memory and a controller device that handles the complexities of error correction and other vendor-specific housekeeping operations associated with using the memory.

News ID 241

Wolfson: S/PDIF transceivers with jitter suppression

Wolfson Microelectronics releases two audio transceivers that offer exceptional jitter rejection. The new WM8804 and WM8805 devices address S/PDIF audio data transfer in LCD televisions, hi-fi, PC motherboards and many other audio systems. The transceivers are structured around an integrated high performance PLL with an intrinsic period jitter of 50ps and jitter rejection frequency of 100Hz.

News ID 244

NatSemi: Power-over-Ethernet powered device controller

National Semiconductor introduces a Power-over-Ethernet powered device controller with adjustable output current level and the ability to interface with any DC-DC converter topology. National's new LM5073 integrates a programmable interface port including a hot-swap controller that exceeds the standard IEEE 802.3af specifications, allowing designers to operate PoE appliances at power levels of 30W or more.

News ID 272

Aicas: partnering with ALT Software to offer OpenGL display capabilities

ALT Software, supplier of DO-178B certifiable 2D/3D graphics solutions and development services announced that it has teamed with aicas, to provide integrated OpenGL drivers for use with aicas Java Virtual Machine for object-oriented software development. This collaboration will yield a flexible, deterministic display software for the aerospace & defence, avionics, automotive/telematics, industrial, consumer, and medical embedded markets.

News ID 369

Advertisers Index

More information about each news story is available on www.embedded-control-europe.com/ece_magazine
You just have to type in the "News ID". —

ARE YOU INTO IT?

Introducing ZiLOG's new ***Z8 Encore! MC™ Flash Microcontroller*** – the perfect solution for motor control applications. Visit us at **www.ZiLOG.com/z8encoremc** to find out how our Z8FMC16100 outperforms the competition. *We're into it because you are.*

Optimized Hardware Peripherals for Faster Time-to-Market – Our unique design ties together key peripherals that work synchronously to free up CPU resources, minimize external components, and eliminate the need for writing complex and time-intensive software algorithms typically found with competing solutions.

Higher Performance/Lower BOM – Integrated peripheral offering developed specifically for motor control, including a 6-channel, 12-bit PWM module with fast shutdown upon overcurrent sensing or multiple fault inputs, a fast 2.5µs A/D converter with a "time stamp" feature, an operational amplifier, a high accuracy internal precision oscillator, a comparator, and multiple communication engines.

Compact Size – 5x5 mm^2, 32-pin QFN package for board-space limited applications.

Vector Control – The only 8-bit MCU to provide 3-phase Vector Control.

Multiple Applications – A solid choice for 3-phase Brushless DC (BLDC), AC Induction, Universal, Switched Reluctance and Brushed DC motor control applications.

Don't compromise your motor control application with less-than-average alternatives. *ZiLOG has the best in class hardware, firmware, and tools to get your product to market faster.*

ZiLOG™

[EMBEDDED CONTROL SOLUTIONS]

WE'RE INTO IT.

FLASH MICROCONTROLLERS | MOTOR CONTROL | INFRARED REMOTE CONTROLS | MICROPROCESSORS | IRDA | MODEMS | SCC

Collect Them All!

Coming Soon to a 430 Day Near You.

Learn More About the World's Lowest Power MCU

TI's **easy-to-use, ultra-low-power MSP430 MCUs** are 16-bit RISC microcontrollers that solve design challenges while saving development time and power. Meet the experts at TI's popular worldwide series of FREE, half-day seminar style demonstrations. Learn how to use the MSP430, and the new **MSP430 Collectors Kit**, which **includes the eZ430-F2013 complete development tool** with removable MSP430F2013 target board, **PLUS the Limited Edition 430 Day Capacitive Touch Board**. Seats and tools are limited. Register today!

To register and for a list of locations, visit www.ti.com/430day-e

You can also enhance your collection by purchasing other eZ430-compatible target boards such as the eZ430-T2012 from Texas Instruments or the MSP-Mojo FIR Filter Board from MSP430 Third Party Quickfilter Technologies to round out your collection or trade with your friends. www.ti.com/430tools

430 Day is brought to you by Texas Instruments and their distribution network.

430 Day 2007

- FREE half-day seminar
- Register and attend to get your FREE MSP430 Collectors Kit, featuring the eZ430-F2013 tool, AND…
- The Limited Edition eZ430-compatible Capacitive Touch Board, available only at 430 Day!

430 Day 2007
Register Today!
www.ti.com/430day-e

1894A0