Telos: Enabling Ultra-Low Power Wireless Research

Joseph Polastre, Robert Szewczyk, and David Culler
Computer Science Department
University of California, Berkeley
Berkeley, CA 94720
{polastre,szewczyk,culler}@cs.berkeley.edu

Abstract— **We present *Telos*, an ultra low power wireless sensor module ("mote") for research and experimentation. Telos is the latest in a line of motes developed by UC Berkeley to enable wireless sensor network (WSN) research. It is a new mote design built from scratch based on experiences with previous mote generations. Telos' new design consists of three major goals to enable experimentation: minimal power consumption, easy to use, and increased software and hardware robustness. We discuss how hardware components are selected and integrated in order to achieve these goals. Using a Texas Instruments MSP430 microcontroller, Chipcon IEEE 802.15.4-compliant radio, and USB, Telos' power profile is almost one-tenth the consumption of previous mote platforms while providing greater performance and throughput. It eliminates programming and support boards, while enabling experimentation with WSNs in both lab, testbed, and deployment settings.**

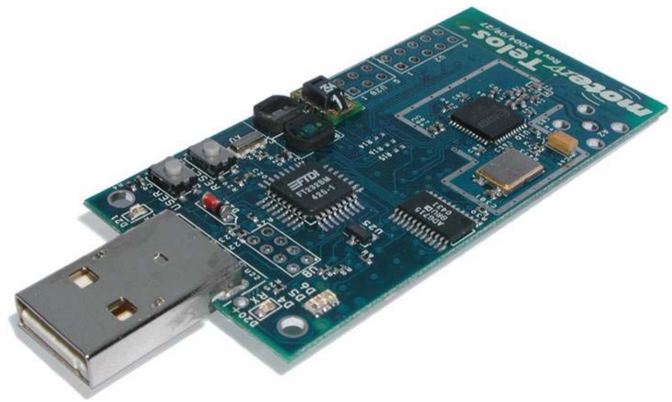

Fig. 1. Telos ultra-low power wireless module ("mote") with IEEE 802.15.4 wireless transceiver.

I. Introduction

Wireless sensor networks are ideally suited for long-lived applications deployed at large densities for low cost. Unfortunately, the current WSN platforms built from commercial off-the-shelf (COTS) components have a lifetime of no more than two years, communicate through non-standard interfaces, are expensive, and are difficult to use for experimentation, development, testing, and deployment.

In this paper, we introduce the design of Telos, the latest wireless sensor device, or "mote", from the University of California, Berkeley. Telos (shown in Figure 1) is a new design to further research in sensor networks with three major goals: lower power operation than previous mote generations, easy to use, and robustness for experimentation and deployment. The Telos design is based on the following low duty cycle principle: the node is asleep for the majority of the time, wakes up quickly on an event, processes, and returns to sleep. For the lowest power consumption, the standby current and wakeup time (time to transition from sleep to active mode) must be minimized [1] since the the active portion of a sensor network application is typically extremely small [2]. Telos offers more than just low power operation through its integrated design. Integration of programming, communication, storage, and sensing allows researchers to utilize more functionality and develop more robust systems.

II. Related Work

The lineage of current platforms can be traced back to a number of devices called "COTS motes" built by the SmartDust project and shown in Figure 2. These devices were built to approximate the capabilities of an envisioned SmartDust node with off the shelf components [3]. These designs used a small 8-bit microcontrollers (4 to 8 kB of flash, 512 bytes of RAM); a simple radio (OOK modulation at 4kbps) and integrated sensors (magnetometers, accelerometers, temperature, pressure, etc). Later designs (weC [4], and René) exposed a custom sensor interface and allowed for the possibility of remote reprogramming.

Mica [5], released in 2001, was carefully designed to serve as a general purpose platform for WSN research. Compared with preceding designs, it offered more memory (4kB of RAM and 128kB of flash), extensive sensor interfaces (8 analog lines, several digital IO channels, dedicated serial busses), and a very flexible radio interface. Mica used the RFM TR1000 and simple modulation techniques. The radio's primitive interfaces allowed low power operation and quick turn-on times. The unbuffered, bit-level radio interface connected to several IO pins, interrupts, and an SPI bus on the main microcontroller; the bus timing was controlled by the CPU clock. Researchers implemented a number of schemes for radio wakeup, low power asynchronous communications, fairly high bandwidth protocols (40 kHz physical layer), and precise time synchronization (to within a 1 bit time).

Mica was useful for development, but unsuitable for deployments. The boost converter provided a stable voltage but used excess quiescent current. The radio communication range was short and relatively unreliable. The extensive I/O connector was not robust to variations in temperature [6]. Mica2, the follow on to the Mica platform, corrected many shortcomings: the boost converter was discarded, and the MCU was replaced with the ATmega128. This lowered the Mica2 standby current to about 17μA, while waking up the system takes up to 4 ms if using the external crystal. The radio transceiver was replaced with the Chipcon CC1000 offering tunable frequencies from 300 to 900 MHz and FSK modulation resilient to noise. The radio exposed a byte-level interface and timing interrupts. Although more resilient, the Mica2 had higher energy per bit and an order of magnitude higher wakeup time. Despite these shortcomings, Mica2 and the smaller Mica2Dot are the *de facto* standard research platforms in WSN research (16 of 21 papers in SenSys 2004 used Mica2 for evaluation). MicaZ [7] continues the evolution of the Mica family: it replaces the CC1000 radio with a CC2420, an IEEE 802.15.4 compatible radio.

A single chip mote implementation called Spec [8] resulted from analyzing the Mica platform. Just 5mm^2 in size, Spec uses a number of dedicated hardware accelerators to perform programmable start

Mote Type Year	WeC 1998	René 1999	René2 2000	Dot 2000	Mica 2001	Mica2Dot 2002	Mica 2 2002	Telos 2004
Microcontroller								
Type	AT90LS8535		ATmega163		ATmega128			TI MSP430
Program memory (KB)	8		16		128			48
RAM (KB)	0.5		1		4			10
Active Power (mW)	15		15		8		33	3
Sleep Power (μW)	45		45		75		75	15
Wakeup Time (μs)	1000		36		180		180	6
Nonvolatile storage								
Chip	24LC256				AT45DB041B			ST M25P80
Connection type	I^2C				SPI			SPI
Size (KB)	32				512			1024
Communication								
Radio	TR1000				TR1000	CC1000		CC2420
Data rate (kbps)	10				40	38.4		250
Modulation type	OOK				ASK	FSK		O-QPSK
Receive Power (mW)	9				12	29		38
Transmit Power at 0dBm (mW)	36				36	42		35
Power Consumption								
Minimum Operation (V)	2.7		2.7		2.7			1.8
Total Active Power (mW)	24				27	44	89	41
Programming and Sensor Interface								
Expansion	none	51-pin	51-pin	none	51-pin	19-pin	51-pin	16-pin
Communication	IEEE 1284 (programming) and RS232 (requires additional hardware)							USB
Integrated Sensors	no	no	no	yes	no	no	no	yes

Fig. 2. The family of Berkeley motes preceeding Telos and their capabilities

symbol detection, bit serialization, and encryption. A simple FSK radio uses a number of unusual structures (e.g. digital frequency-lock-loops) to reduce startup times and active power, while still providing the frequency agility and improved resistance to noise. The CPU has been optimized to reduce the cost of context switching. Spec's performance is over 1000 times better than Mica in many applications. Unlike the Mica family, Spec is fully integrated and offers limited interface flexibility. Since it is a research project, it is unlikely to become available in quantities to the research community.

Spec provides significant advantages in power consumption due to its integrated design and hardware accelerators. The Telos design parallels that of Spec– instead of integrating the design into silicon, Telos uses COTS components with hardware accelerators to build a power efficient system that does not sacrifice performance.

III. Telos Platform

We discuss the design and implementation of Telos, including the intuition behind hardware selection. We offer an analysis of Telos and existing devices and provide a discussion of research enabled by the Telos platform.

The Telos platform is the result of over 12 months of research and development by two full-time graduate students, and numerous collaborator, at the University of California, Berkeley. Telos is a completely new design based off experiences from using previous mote generations designed by former graduate students at Berkeley and researchers at other institutions (Intel, ETH Zurich, etc...). We discuss how we achieve the three major goals for Telos: ultra-low power operation, easy to use, and robust hardware and software implementation.

A. Technological Trends

Since the Mica2 mote was released in 2002, a number of new microcontrollers have been introduced offering lower power consumption, more on-chip peripherals, and various RAM and flash sizes. Our low power principle focuses on reducing the sleep current and wakeup time of our system. Figure 4 summarizes the microcontroller improvements over time.

For Telos, we chose the MSP430 microcontroller after evaluating existing products from Atmel, Motorola, and Microchip. Figure 4 shows that the MSP430 has the lowest power consumption in sleep and active modes. The microcontroller operates down to 1.8V. Low voltages are important for extracting all of the energy out of a power source–e.g., AA batteries have a cut-off voltage of 0.9V. If two batteries are used in series, the system cut-off voltage is 1.8V, exactly the same as the minimum required voltage for the MSP430. In contrast, the ATmega128 MCU (Mica family) will only run down to 2.7V, leaving almost 50% of the AA batteries unused. The MSP430 also has the fastest wakeup time of any microcontroller; transitioning from standby (1μA) to active mode (8MHz) in no more than 6μs. The MSP430 has a DMA controller to reduce load from the MCU core, lower power consumption, and increase performance.

The trend is to keep the RAM and Flash sizes constant (shown in Figure 4) while adding additional hardware accelerator modules. The MSP430 provides us with the largest on-chip RAM buffer (10kB), useful for on-chip signal processing. Larger RAM buffers enable more sophisticated applications–for example, Maté [9] can use extended RAM to support more execution contexts and larger program images; TinyDB [10] uses larger RAM storage for in-network aggregation and data table storage. Larger flash storage, although useful for large applications, has not been the limiting factor in developing WSN applications to date.

There are two distinct types of low power, low data rate radios: narrowband and wideband radios as shown in Figure 3. Many narrowband radios provide very fast startup times since they are clocked by the MCU but have simple modulation schemes, no spreading codes, and are not robust to noise. Wideband radios must wait for high speed oscillators to start. Enhanced modulation schemes found in wideband radios, such as spread spectrum (DSSS) and phase shift keying (O-QPSK), provide signal robustness to noise and interference. Narrowband radios typically operate at lower frequencies and lower data rates; wideband radios typically operate in the 2.4GHz band and provide higher data rates. To pick the most applicable radio, we must evaluate the impact of noise, flexibility available to the end application, ease of communication with other devices, power

Type	Narrowband				Wideband		
Vendor Part no.	RFM TR1000	Chipcon CC1000	Chipcon CC2400	Nordic nRF2401	Chipcon CC2420	Motorola MC13191/92	Zeevo ZV4002
Max Data rate (kbps)	115.2	76.8	1000	1000	250	250	723.2
RX power (mA)	3.8	9.6	24	18 (25)	19.7	37(42)	65
TX power (mA/dBm)	12 / 1.5	16.5 / 10	19 / 0	13 / 0	17.4 / 0	34(30)/ 0	65 / 0
Powerdown power (μA)	1	1	1.5	0.4	1	1	140
Turn on time (ms)	0.02	2	1.13	3	0.58	20	*
Modulation	OOK/ASK	FSK	FSK,GFSK	GFSK	DSSS-O-QPSK	DSSS-O-QPSK	FHSS-GFSK
Packet detection	no	no	programmable	yes	yes	yes	yes
Address decoding	no	no	no	yes	yes	yes	yes
Encryption support	no	no	no	no	128-bit AES	no	128-bit SC
Error detection	no	no	yes	yes	yes	yes	yes
Error correction	no	no	no	no	yes	yes	yes
Acknowledgments	no	no	no	no	yes	yes	yes
Interface	bit	byte	packet/byte	packet/byte	packet/byte	packet/byte	packet
Buffering (bytes)	no	1	32	16	128	133	yes *
Time-sync	bit	SFD/byte	SFD/packet	packet	SFD	SFD	Bluetooth
Localization	RSSI	RSSI	RSSI	no	RSSI/LQI	RSSI/LQI	RSSI

* Manufacturer's documentation does not include additional information.

Fig. 3. Capabilities of current COTS radios suitable for WSNs, their features, and power profile.

Manufacturer	**Device**	**RAM (kB)**	**Flash (kB)**	**Active (mA)**	**Sleep (μA)**	**Release**
Atmel	AT90LS8535	0.5	8	5	15	1998
	Mega128	4	128	8	20	2001
	Mega165/325/645	4	64	2.5	2	2004
General Instruments	PIC	0.025	0.5	19	1	1975
Microchip	PIC Modern	4	128	2.2	1	2002
Intel	4004 4-bit	0.625	4	30	N/A	1971
	8051 8-bit Classic	0.5	32	30	5	1995
	8051 16-bit	1	16	45	10	1996
Philips	80C51 16-bit	2	60	15	3	2000
Motorola	HC05	0.5	32	6.6	90	1988
	HC08	2	32	8	100	1993
	HCS08	4	60	6.5	1	2003
Texas Instruments	TSS400 4-bit	0.03	1	15	12	1974
	MSP430F14x 16-bit	2	60	1.5	1	2000
	MSP430F16x 16-bit	10	48	2	1	2004
Atmel	AT91 ARM Thumb	256	1024	38	160	2004
Intel	XScale PXA27X	256	N/A	39	574	2004

Fig. 4. Microcontroller history: The main table contains traditional microcontrollers; the bottom two devices are 32-bit microprocessors presented for comparison.

consumption, startup times, and available data bandwidth. Figure 3 provides a summary of common radio features. No single radio in Figure 3 is globally optimal; instead a radio must be chosen based on application requirements.

For Telos, we chose to use the new IEEE 802.15.4 standard. By using a standardized radio, Telos can communicate with any number of devices sharing the same physical layer, including devices from other vendors. The primary factor in selecting a radio technology was the desire experiment with the new IEEE 802.15.4 wireless technology. Telos uses the Chipcon CC2420 radio in the 2.4GHz band, a wideband radio with O-QPSK modulation with DSSS at 250kbps. The higher data rate allows shorter active periods further reducing energy consumption. The radio crystal used on Telos was carefully chosen to be a low-ESR 16MHz crystal; low resistance is essential for minimizing the startup time of the crystal (and thus minimizing wasted energy), The Telos crystal yields a 580μs startup time, almost 300μs lower than the minimum advertised startup time by Chipcon.

Since IEEE 802.15.4 radio interfaces are packet-based, we lose considerable flexibility in software for controlling the radio's operation. The CC2420 provides a number of hardware accelerators to achieve better performance. These include encryption and authentication, packet handling support, auto acknowledgments, and address decoding. Since hardware accelerators are embedded in the radio instead of the microcontroller, the accelerators may not be used as general purpose functions. For example, a data buffer may be encrypted and stored in flash, however since it is not being sent over the radio, the radio's hardware encryption module cannot be used. Other downsides include auto acknowledgment support–when this feature is used, packets not addressed to the local node are discarded by hardware preventing services from overhearing messages useful for link estimation and routing.

B. Integrated Design

Instead of using separate pluggable modules to create a full sensor node, Telos integrates programming, computation, communication, and sensing onto a single device. The integrated design provides an easy to use mote with increased robustness.

Telos uses an internal 2.4GHz Planar Inverted Folded Antenna (PIFA) built into the printed circuit board and tuned to match the radio circuitry. An optional SMA coax connection may be used instead of the internal antenna. Integration of the antenna lowers the overall cost of the mote since no expensive external antennae are needed.

Telos is programmed (either with the bootstrap loader or JTAG) through on-board USB that also provides power. USB was chosen since it is a standardized protocol that interfaces well with current PCs. On-board USB is extremely easy to use and has lowered development time compared to non-standard mote interfaces.

Telos has a user button, reset button, and 16-pin IDC expansion header. The user and reset button signals are exported via the header so the physical user-interface may be located on support hardware. The reset button may be retasked as a non-maskable interrupt allowing it to be used as a power button instead. By exporting I^2C and UART over the expansion header, I/O bus expanders can be used to attach as many connections as are found on legacy "Mica-style" sensor boards [11].

Inside the Telos, it is the first mote to include hardware write-protection for external storage. When plugged into a USB port, the write protection is disabled and the first segment of the external flash may be written. When running on batteries (without USB), the segment is write protected. Hardware write protection is essential for systems that may be reprogrammed wirelessly–since a known good program image may be stored in the write protected flash, there is always a fallback mechanism to a usable mote.

Each hardware "sub-circuit" is isolated; power to the circuit can be turned on or off independently of the rest of the platform. This

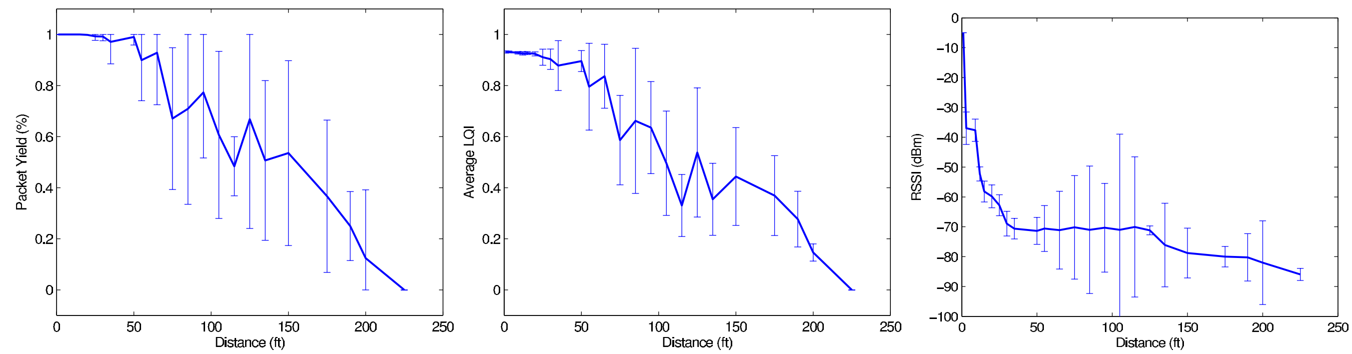

Fig. 5. Packet yield (*left*), link quality indicator (LQI,*center*), and received signal strength (RSSI,*right*) outdoors with the Telos mote and internal antenna. The results are averaged over 10 receivers co-located. From 75-110 feet, a dip in the terrain yields more erratic readings and wider variation in RSSI.

isolation provides a degree of robustness–in the event of a failure, faulty modules can be disabled to minimize their impact on the system. The motivation for this design comes from the experience with real-world sensor networks on Great Duck Island (GDI) [12], [6]. On GDI, one of the main predictors of node failure was the existence of a failed sensor. Since the failure can be recognized in software, the ability to cut power to that section of the board may have saved the system as a whole.

Since the IEEE 802.15.4 protocol has a 64-bit addressing scheme, we have included a 48-bit silicon serial identification chip. The id, combined with a manufacturer's IEEE Organizationally Unique Identifier (OUI) stored in write protected flash, provides the user with a valid, unique 64-bit MAC address. The MAC address is useful for system and network diagnostics, as well as absolute node identification.

C. Analysis

Our analysis of the Telos platform focuses on the platform's power consumption and the features that further research in sensor networks. The power consumption of a sensor module is not just the microcontroller and/or radio, but also the auxiliary components and their quiescent current. The power consumption of the Telos mote for various operations compared to the existing Mica2 and MicaZ platforms is shown in Figure 6.

Telos features a lower power flash and microcontroller than Mica2 (Atmel with CC1000 radio) and MicaZ (Atmel with CC2420 radio). Due to Telos' integrated design, 3μA additional current in sleep state is sacrificed to switches and buffers that protect current from flowing backward into disconnected components, specifically the USB circuitry. Despite this sacrifice, the overall power consumption of a sampling cycle (wakeup, sample, transmit, and sleep) is lower than existing platforms. The power consumption is the total time the mote is active multiplied by the current consumption during that time. Since Telos has lower current consumption, lower startup time, and lower operating voltage for the entire mote, it can achieve longer lifetimes than previous designs. At a 1% duty cycle, Telos can last for almost 3 years. For comparison, the lifetime of the Mica2 mote is 1.5 years and the MicaZ mote is 1 year [1].

Lower power consumption does not imply that Telos has less functionality. Powerful microprocessor modules are now being integrated into embedded microcontrollers. Telos features a DMA controller that operates while the MCU core is sleeping. The DMA permits applications to sample from the ADC, output a voltage on the DAC,

Operation	Telos	Mica2	MicaZ
Minimum Voltage	1.8V	2.7V	2.7V
Mote Standby (RTC on)	5.1 μA	19.0 μA	27.0 μA
MCU Idle (DCO on)	54.5 μA	3.2 mA	3.2 mA
MCU Active	1.8 mA	8.0 mA	8.0 mA
MCU + Radio RX	21.8 mA	15.1 mA	23.3 mA
MCU + Radio TX (0dBm)	19.5 mA	25.4 mA	21.0 mA
MCU + Flash Read	4.1 mA	9.4 mA	9.4 mA
MCU + Flash Write	15.1 mA	21.6 mA	21.6 mA
MCU Wakeup	6 μs	180 μs	180 μs
Radio Wakeup	580 μs	1800 μs	860 μs

Fig. 6. Measured current consumption of Telos compared to Mica2 and MicaZ motes

or transfer data to and from the radio without interrupting the MCU. DMA is traditionally used to increase performance, but in the case of low power embedded systems, the DMA controller actually lowers the duty cycle by permitting the MCU core to remain asleep longer and service less interrupts. The performance enhancements of DMA permit up to 200ksamples/sec ADC sampling, compared to a maximum of 10ksamples/sec on MCUs without DMA (some powerful MCUs can achieve as high as 70ksamples/sec, but no interrupt-based method on current MCUs can achieve 200ksamples/sec). The lower sampling rate is caused by the overhead of context switching due to interrupts after each ADC conversation. Telos also has built-in brown-out reset, supply voltage supervisor, and interrupt driven sensors to maximize the sleeping time of the mote.

For communications, IEEE 802.15.4 radios provide applications with information about the incoming signal. Telos' on-board antenna provides a mostly omnidirectional pattern[1]. We tested the effect of distance on received signal strength (RSSI), packet success rate, and link quality (LQI). LQI is a metric introduced in 802.15.4 that measures the error in the incoming modulation of successfully received packets (packets that pass CRC check). We placed 10 receivers at the same location and 1 node transmitting at 0dBm at a distance *d*, all located 4" above the ground outdoors. We averaged the results over all receivers. Figure 5 shows the average packet success, LQI, and RSSI for Telos using the internal antenna. The LQI provided by the radio closely maps to the packet success rate as shown in Figure 5. The RSSI follows an exponential decrease while the packet success rate is high; after 60 feet, the signal is noisier and decreases to the minimum sensitivity of the transceiver. The small variance in

[1]More microbenchmarks including radio and antenna impedance matching can be found in the Telos datasheet [13].

RSSI among receivers and the correlation between LQI and packet success rate provides additional information not previously available to network services such as multihop networking and localization.

The consequence of using a higher speed radio is that it may saturate the MCU's processing capabilities when the channel is fully loaded. We ran experiments on a 30 node Telos network to measure the effective bandwidth. A single Telos node is able to source approximately 1/2 of the full data bandwidth of the channel, or 125kbps. When all 30 nodes transmit as quickly as possible, Telos is limited to an average reception rate of 150kbps. Our current implementation is interrupt driven; however we intend to increase performance by using the DMA controller to directly transfer data from the radio, reduce the number of interrupt context switches, and reduce the number of receive buffer overflow events.

D. Enabling Research

To support current research efforts, Telos integrates a number of features that create more robust systems. Deluge [14] is an epidemic code-propagation protocol used in BNP [15] to reprogram the network wirelessly. In the event that a faulty code image renders the node unusable, a halted node may be reset via the watchdog timer or a button (referred to as a "Golden Gesture"). Telos will automatically reload the microcontroller's code flash with the hardware protected golden image.

The golden gesture can be performed through a number of sequences. Since Telos features a "User Button" for external user input, a combination of pressing the reset and user buttons may be required to boot to the golden image. The user button may serve other user programmable services thereby providing a method for physical input to be received at the mote when radio commands are not an option (such as during physical deployment).

Due to Telos' low wake up times, the mote is automatically put to sleep when there is no active processing. By automatically entering the lowest power mode when idle, Telos has a lower operating power profile. The low power profile also makes Telos attractive for rechargeable designs, including solar and vibration harvesting. Since Telos operates down to 1.8V, super-capacitor designs are now possible (many super-capacitors operate at a maximum of 2.5V, lower than the minimum operating voltage of previous motes). Since the sleep current and wakeup times are lower, harvested energy may be stored quicker and be used for more operations.

Finally, Telos lowers the barrier of entry for using WSNs. By using USB on every mote, any Telos may operate as an experimental device on a lab bench, a gateway to a PC or higher functionality device, or as a node in a large testbed. In the lab or classroom, USB provides an easy and robust way to interface, program, and experiment with motes. As a gateway, no programming or interface board is required and any node may act as the gateway. Finally, off-the-shelf USB products are a low cost method for deploying testbeds with a back-channel link. Back-channels are important for developing algorithms on the motes while being able to debug their state and operation without relying on the wireless transceiver. Most testbeds with back-channel links are created using long runs of Ethernet. The infrastructure cost of a 60-node Telos testbed is approximately $1,000 ($600 in cabling, $400 in USB hubs). For longer cable runs, a Linksys Ethernet to USB device may be purchased for $79 and includes 2 USB ports. A networked testbed with Linksys devices is an additional $2,400. In contrast, a 60-node Mica2 testbed costs almost $21,000 in infrastructure hardware alone ($349 per Mica2 Ethernet adapter as of February 2005).

E. Software Implications

There is a huge impact on the software when creating a new hardware platform. TinyOS [16] is a componentized operating system suitable for research in wireless embedded systems. The composition of components and whole program analysis allows researchers to work on the system at any level (e.g., the details of link protocols up to the application semantics). Since the MSP430 is a different architecture than the microcontrollers commonly used in TinyOS, we were forced to rethink hardware abstraction for embedded systems. An opportunity was created by Telos to redesign the existing TinyOS 1.1.x interfaces to create effective abstractions that take advantage of the powerful hardware features of the MSP430 microcontroller.

We developed a three-tier architecture to provide a hardware independent abstraction regardless of microcontroller or radio. The design is described in detail by Handziski et. al. [17]. We chose to expose the primitives of the hardware, such as register access and module flags, through a hardware presentation layer (HPL). A platform-dependent hardware abstraction layer (HAL) exposes hardware module functionality so that the full power of the hardware may be used. The HAL includes getting data from the ADC, setting a hardware Timer, or writing to external flash. The hardware independent layer (HIL) exposes a subset of a platform's capabilities that are available to system services. The HPL/HAL/HIL model is implemented in TinyOS for the TI MSP430 microcontrollers and has been adopted as the basic architecture for hardware abstraction by the TinyOS 2.0 Working Group.

On top of the HAL/HPL/HIL abstraction, we built a platform-independent radio stack (link protocol and physical layer access) for the CC2420 transceiver. Each platform using the CC2420 implements a set of components that provide register access to the radio; the radio stack then acts as a library that uses these primitives to the control the radio. Our CC2420 implementation operates on Telos, MicaZ, iMote2, and Chipcon CC2420EB platforms. Since these platforms all share the same physical layer, TinyOS enables cross-platform communication and research on hybrid networks.

IV. Conclusion

We have presented the design and implementation of Telos, the latest generation in a family of motes from UC Berkeley. We showed that Telos is the lowest power mote to date. Telos includes numerous enhancements that enable research in wireless sensor networks while making the devices easier to use and lowering the per-module cost. Other features, like hardware write protection and radio signal stability, closely map to current research. Researchers may experiment with the new IEEE 802.15.4 standard and use existing work in TinyOS. Additional flexibility allows software to configure or disable hardware modules. Telos is a robust module with lower power consumption yet greater performance than existing designs.

References

[1] J. Polastre, R. Szewczyk, C. Sharp, and D. Culler, "The mote revolution: Low power wireless sensor networks," in *Proceedings of the 16th Symposium on High Performance Chips (HotChips)*, Aug. 2004.

[2] V. Shnayder, M. Hempstead, B. rong Chen, G. Werner-Allen, , and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proceedings of the Second ACM Conference on Embedded Networked Systems (SenSys)*, Nov. 2004.

[3] S. Hollar, "Cots dust," Master's thesis, University of California, Berkeley, 2000.

[4] J. McLurkin, "Algorithms for distributed sensor networks," Master's thesis, University of California, Berkeley, 1999.

[5] J. Hill and D. Culler, "Mica: a wireless platform for deeply embedded networks," *IEEE Micro*, vol. 22, no. 6, pp. 12–24, November/December 2002.
[6] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler, "Lessons from a sensor network expedition," in *Proceedings of the First European Workshop on Wireless Sensor Networks (EWSN)*, Jan. 2004.
[7] Crossbow Technology Inc., "MICAz wireless measurement system," http://www.xbow.com, June 2004.
[8] J. Hill, "System architecture for wireless sensor networks," Ph.D. dissertation, University of California at Berkeley, 2003.
[9] P. Levis and D. Culler, "Mate: A tiny virtual machine for sensor networks," in *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2002.
[10] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "The design of an acquisitional query processor for sensor networks," in *Proceedings of SIGMOD*, June 2003.
[11] J. Polastre, "Interfacing Telos to 51-pin sensorboards," http://www.tinyos.net/hardware/telos/telos-legacy-adapter.pdf, Oct. 2004.
[12] R. Szewczyk, A. Mainwaring, J. Polastre, and D. Culler, "An analysis of a large scale habitat monitoring application," in *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Nov. 2004.
[13] Moteiv Corporation, "Telos (Rev B) Datasheet," http://www.moteiv.com, Dec. 2004.
[14] J. W. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," in *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Nov. 2004.
[15] J. Hui, "TinyOS network programming (version 1.0)," TinyOS 1.1.8 Documentation, Aug. 2004.
[16] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "TinyOS: An operating system for wireless sensor networks," in *Ambient Intelligence*. Springer-Verlag, 2005.
[17] V. Handziski, J. Polastre, J.-H. Hauer, C. Sharp, A. Wolisz, and D. Culler, "Flexible hardware abstraction for wireless sensor networks," in *Proceedings of the Second European Workshop on Wireless Sensor Networks (EWSN)*, Feb. 2005.