

# EcoSpire: An Application Development Kit for an Ultra-Compact Wireless Sensing System

Chien-Ying Chen, Yu-Ting Chen, Yi-Hsuan Tu, Shun-Yao Yang, and Pai H. Chou

**Abstract**—EcoSpire is a hardware and software platform for wireless sensing applications. The hardware kit consists of two types of self-contained, expandable sensor nodes, as well as, a multipurpose base station and programmer, a cascadable charger, and sensor modules. The software kit consists of an integrated development environment (IDE) for application programming and RF debugging utilities on the host computer, plus a lightweight operating system with support for data logging and remote firmware reprogramming. Together, this package facilitates application development of ultra-compact, wearable wireless sensing systems.

**Index Terms**—Embedded systems, integrated development environment, runtime support, platform-based design, wireless sensor networks.

## I. INTRODUCTION

WIRELESS Sensor Networks (WSN) are finding use in many industrial, scientific, and consumer applications in recent years. The applications have diverse requirements in terms of physical size, computing capabilities, wireless bandwidth, and real time latency. A custom design for each application would require much duplicated effort each time, whereas a reusable *platform* enables developers to work mainly at the application level. However, the generality often comes at a price. Most *nodes* (wireless sensor nodes) today are nowhere near the “smart dust,” as the software complexity forces the use of components with larger memory capacity or more I/O pins than users would like. To reduce complexity, researchers often replace industry-standard networking stack with their own. This not only defeats interoperability, but also prevents them from taking advantage of faster, more energy-efficient, more robust, or more integrated radio designs.

Our goal is to fill gaps left by current platforms particularly in the ultra-compact, low cost, high data rate space. Although we previously described an early version of the hardware [1], this letter contains not only an entirely new design, but more importantly, a complete development kit consisting of both the

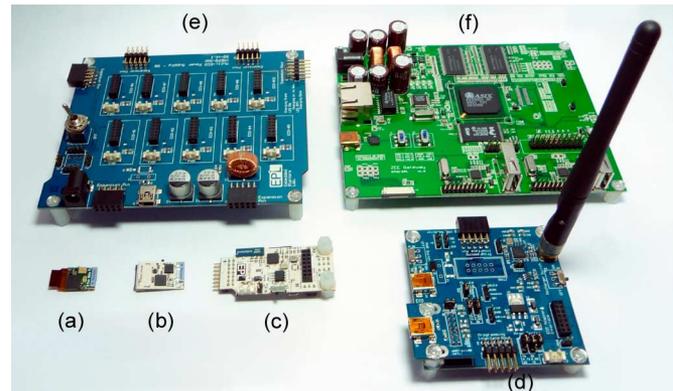


Fig. 1. Photograph of EcoSpire development kit. (a) Eco node (b) EcoSpire simple node (c) EcoSpire super node (d) Development board (e) Charger (f) EZ-Gate gateway.

hardware/software platform, infrastructure support, and a tool chain. Our contributions are as follows.

- 1) New EcoSpire *simple node* and *super node* [Fig. 1(b), (c)] with 2 Mbps of data rate, plus expansion modules for sensors, actuators, storage, radio, and power.
- 2) Supporting hardware, including multipurpose *Development Board*, *EZ-Gate gateway* that not only bridges with TCP/IP, but also supports heterogeneous wireless interoperability, and a multibattery charger [Fig. 1(d)(f)(e)].
- 3) *Enix* lightweight operating system that supports concurrent programming with low complexity on the nodes.
- 4) An integrated development environment (IDE) with the compiler tool chain, system development utilities, an API library, and debugging tools.

Together, the collection of platform and tools is called the EcoSpire kit. This letter describes the EcoSpire kit except for the gateway software [2].

## II. RELATED WORK

Wireless sensor platforms proposed to date differ in the type and capability of microcontroller unit (MCU), external storage, the radio transceiver, the power sources, and many more. Table I shows a comparison of some related WSN nodes.

One main decision is whether to use separate or integrated MCU and RF components. Most nodes [3]–[6] use separate ones. Even though integrated ones are available, they have less built-in program memory (16–64 KB) than those similar sized RF-less MCUs (64–128 KB) do. The latter better matches the requirements of the typical runtime system and protocol stack today, but even at the same capacity, the latter is always physically larger due to less on-chip integration.

Manuscript received November 11, 2009; revised November 15, 2009. First published December 04, 2009; current version published February 05, 2010. This manuscript was recommended for publication by R. Kastner. This work was supported in part by the National Science Foundation CAREER Grant CNS-0448668, CNS-0721926, the National Science Council (Taiwan) Grant NSC 96-2218-E-007-009, and Ministry of Economy (Taiwan) Grant 96-EC-17-A-04-S1-044.

C.-Y. Chen, Y.-T. Chen, Y.-H. Tu, and S.-Y. Yang are with the National Tsing Hua University, Hsinchu City, Taiwan 30013 (e-mail: dogs007.tw@gmail.com; wtg.design@gmail.com; cindyduh@gmail.com; asdyaya@gmail.com).

P. H. Chou is with the University of California, Irvine, CA 92697 USA (e-mail: pai.chou@gmail.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/LES.2009.2037984

TABLE I  
COMPARISON AMONG COMMON SENSOR PLATFORMS

	EcoSpire simple node	EcoSpire super node	Mica2 Dot	MicaZ OEM *	Tmote mini *	ZN1	EPIC core *
Size	2x1.3x0.2 cm	5x2.3x0.2 cm	2.5x2.5x0.6 cm	2.4x2.4x0.5 cm	2.5x1.5x0.5 cm	1.5x1.5x0.4 cm	2.4x2.4x0.2 cm
Microcontroller							
Type	nRF24LE1	nRF24LE1	ATmega128L	ATmega128L	MSP430	H8S/2218	MSP430
Frequency (MHz)	16	16	8	8	8	24	8
Data mem. (KB)	1	1	4	4	10	12	10
Code mem. (KB)	16	16	128	128	48	128	48
External Storage							
Ext. mem.	no	SD (2GB)	Flash (512KB)	Flash (512KB)	EEPROM (1MB) and SD	no	Flash (2MB)
Communication							
Radio	nRF24L01 (integrated)	nRF24L01 (integrated)	CC1000	CC2420	CC2420	CC2420	CC2420
Data rate	2 Mbps	2 Mbps	38.4 Kbps	250 Kbps	250 Kbps	250 Kbps	250 Kbps
Protocol	Enhanced Shock Burst	Enhanced Shock Burst	SmartRF	IEEE 802.15.4**	802.15.4	802.15.4	802.15.4
Antenna	Built-in Chip	Built-in Chip or Ext	External	External	External	External	Built-in trace or ext
Power source							
Type	Li-polymer	Li-polymer	Coin Cell	external	external	CR2	external
Capacity (mAh)	80 (replaceable)	80 (replaceable)	540	n/a	n/a	800	n/a
In-System Recharge	yes	yes	no	no	no	no	no
Others							
Sensors	On-board accel. + Separate module	On-board accel. + Separate module	On-board temp. + Separate module	Separate module	Separate module	On-board Sensors + Separate module	Separate module
Host & prog. interface	USB or UART	USB or UART	UART	UART	USB	USB	USB

\* MicaZ OEM, Tmote mini, and EPIC core are meant to be integrated into another large design and are expected to draw power from some external sources.

\*\* IEEE 802.15.4 is the MAC layer below the ZigBee network stack and is often labeled “ZigBee ready,” but in practice the ZigBee stack takes up  $\geq 64$ –96 KB of code size.

Several WSN platforms [3]–[5], [7] contain no on-board sensors. They require sensor modules to be connected externally, or they may function as relay-only nodes. Those with on-board sensors can be made more compact and more reliable, but being a platform means an expansion connector may still be required. Flex-PCB connectors [1] and pads on the edge of the PCBs that fit in standard leadless chip carriers [8] have been proposed as ways to minimize the size of the connector.

Virtually all nodes rely on batteries as the primary power source. AA or AAA are widely available and thus, are the most common for WSN platforms, but they may be too large and heavy for wearable applications. Most nodes provide no support for in-system recharging. ZN1 [9] and Mica2DOT [3] use coin-cell batteries, which have good energy density, but low power density and are not rechargeable. Eco and EcoSpire support in-system recharge.

### III. HARDWARE

The hardware kit consists of two types of nodes, expansion modules, development board, charger, and EZ-Gate gateway.

#### A. Super Node and Simple Node

Two versions of EcoSpire hardware are the *simple node* and *super node*. As shown in Fig. 2, their respective physical dimensions including the rechargeable battery are  $13 \times 20 \times 8 \text{ mm}^3$  and  $23 \times 50 \times 8 \text{ mm}^3$ . The super node contains everything on the simple node, plus a micro-SD slot and a 32-Mbit on-board flash.

1) *MCU and Radio*: EcoSpire uses the Nordic nRF24LE1 system-on-chip, which contains an 8052-compatible MCU core, an nRF24L01 transceiver, a 16 KB program flash, 1 KB data RAM, 12-bit ADC, SPI, I<sup>2</sup>C, and UART. The radio transceiver implements the Enhanced Shockburst protocol in the 2.4 GHz ISM band with 2 Mbps of data rate, 8 times that of IEEE 802.15.4 at the same power level in the same band. The

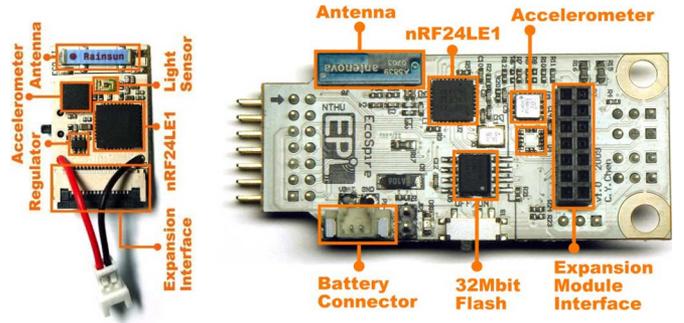


Fig. 2. Photograph of EcoSpire simple node and super node.

hardware supports autoack, autoretransmission, and AES-128 encryption to further facilitate reliable, secure communication.

2) *On-Board Sensors and Expansion Modules*: Both the super node and simple node have an on-board triaxial accelerometer. The super node uses the LIS331DL with digital output over SPI. It has a  $\pm 8 \text{ g}$  range, consumes less than 1 mW, and supports programmable threshold detection, enabling the MCU to sleep during physical inactivity. The simple node uses the ADXL335 ( $\pm 3 \text{ g}$ ) with analog output. Other sensors with digital or analog output can also be added via the expansion module interface, which includes both power input and output. In addition, several modules have been developed, from buttons, microphone, speaker, and LCD to GPS, gyroscope, and camera. Designers can also evaluate built-in serial flash versus external storage media. The super node not only has a built-in chip antenna, but also a connector for an external one, enabling the user to make size versus gain tradeoffs.

3) *Power*: EcoSpire uses the LTC3410 buck regulator with an input voltage range of 2.5–5.5 V. Its maximum output current is 300 mA, for the node and its expansion module. Both EcoSpire nodes use a rechargeable Li-ion battery with a voltage range of 3.7–4.2 V. EcoSpire hardware can monitor the charge level with a built-in fuel gauge for power management.

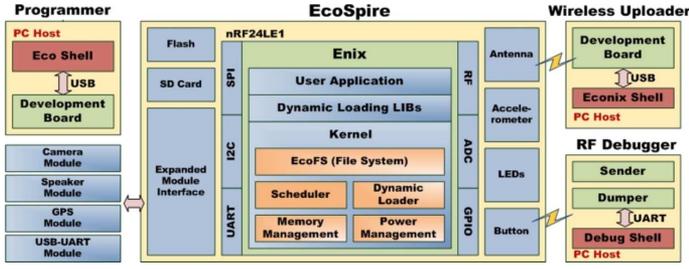


Fig. 3. Block diagram of EcoSpire kit.

TABLE II  
COMPARISON BETWEEN ENIX AND TINYOS

OS	Code size	Binary structure	Programming paradigm	Interactive shell	File system	Remote reprogram module
Enix	8138 B	modular	Cooperative thread	EcoShell	EcoFS	Yes
TinyOS [10]	20924 B <sup>1</sup>	monolithic	Event driven	No	Yes <sup>2</sup>	Yes <sup>3</sup>

<sup>1</sup> It is the code size of TinyOS v1.1.15, where various sensor drivers, as well as network module are included, but excluding storage system and remote reprogramming.

<sup>2</sup> via Capsule (A TinyOS 1.x-based Sensor Storage system).

<sup>3</sup> via XNP (TinyOS v1, single-hop) or Deluge (TinyOS v2, multihop).

## B. Supporting Hardware

1) *Development Board*: The development board can be used as a programmer to write firmware to Eco or EcoSpire nodes, and as a USB base station. Multiple boards can be cascaded via I<sup>2</sup>C, SPI, or UART for parallel processing.

2) *Charger*: Each charger board can charge up to 10 batteries directly or while powering Eco nodes. The boards can be cascaded horizontally and vertically to charge several hundred nodes at a time.

3) *EZ-Gate Gateway*: EZ-Gate is a Linux-based gateway that bridges between WSNs and TCP/IP over Fast Ethernet. A distinguishing feature is that it supports not only the full ZigBee coordinator stack based on the RadioPulse LM-2400 module, but also EcoSpire via a pair of nRF24L01 transceivers. Multiple EZ-Gates support handoff for node mobility. They also support ZigBee interoperability by creating virtual identities for EcoSpire nodes in ZigBee space [2].

## IV. SOFTWARE AND TOOLS

The software for EcoSpire includes the Enix operating system (OS) on the sensor node and the base station, IDE and graphical user interface tools (GUI) on the host computer, and utility tools for uploading images and RF debugging (as shown in Fig. 3).

### A. Enix Operating System

A lightweight OS called Enix runs on EcoSpire. Unlike the popular event-driven TinyOS [11], which can be viewed as run-time-support routines that are linked with the rest of the application as a monolithic executable, Enix is modular and supports dynamic loading. Table II shows a comparison.

1) *Compiler-Assisted Demand Segmentation and Runtime Reprogramming*: Each library function is modified at compile time to be position-independent code and is loaded on demand. This enables code to be swapped in from external flash

memory. A wireless code image update manager and a dynamic loader are also included to enable in-field reprogramming of nodes over RF. TinyOS by default would perform full image replacement, whereas Enix updates only necessary modules.

2) *Dual-Mode Operation and Cooperative Threading*: Instead of executing a monolithic binary image, Enix supports dual-mode operation, where the OS kernel and user program are separated. Enix supports cooperative threads to facilitate expression of concurrency in user programs.

3) *Device Drivers*: Included are hardware drivers for RF, ADC, I<sup>2</sup>C, SPI, external flash, SD card, and accelerometer.

4) *EcoFS File System*: Enix implements a file system called EcoFS using SD card as the storage medium. Instead of treating flash memory as raw pages, it supports data types such as code, preferences (name-value pairs), sensed data (sequenced and time stamped), and network state. When used in conjunction with the compiler-assisted Enix pager, EcoSpire can run with a much larger code space than its physical size.

## B. Tools

1) *Interactive Shell*: A utility shell that runs on the host computer can be used to access and configure the SD card with EcoFS data. It also provides tools for viewing logged data as time history, frequency domain, histogram, etc.

2) *Integrated Development Environment (IDE)*: The EcoSpire IDE contains a complete tool chain and libraries to support application programming. Our IDE is implemented as a plug-in for the Eclipse open source IDE. It supports source code editing, compiler invocation, and firmware programming. Future versions will include a simulator plug-in. Some of our users without any WSN background were able to construct a basic WSN application program in about five minutes.

3) *RF Debugging Tools*: RF interfaces can be difficult to debug due to both the inherent lossy nature of the wireless medium and the several abstraction levels in which errors may occur. Our RF tool complements equipment such as spectrum analyzers by supporting higher level concepts, such as header decoding and calculating CRC specific to each protocol. Our RF tool can also assist testing and debugging by emulating the behavior of a sensor node as either sender or receiver, and by scanning network addresses in a given range to discover the available nodes in the network.

## C. Reference Applications

Several reference applications are included with the kit for developers to modify and extend as a quick way to develop a working system.

1) *EcoNet Smart Home Network System*: EcoSpire nodes are programmed to read and transmit sensor data for temperature, light, and air quality in a home. We have also prototyped nodes for controlling home appliances. The gateway enables the user to manage an EcoNet via a PC or cell phone on the same LAN or remotely. When users add or remove nodes, the nodes adjust their network connection accordingly.

2) *Infant and Child Monitoring*: EcoSpire nodes have been programmed to use the built-in sensor to monitor the temperature and motion of infants. In addition, a camera module has been attached to an EcoSpire node so that it can take snapshots

TABLE III  
POWER COMPARISON: ECOSPIRE VERSUS MICAZ VERSUS TMOTE

	Mode	EcoSpire	MICAz <sup>a</sup>	Tmote <sup>a</sup>
Processor	Active	4mA @16MHz @3V	8mA @8MHz @3V	2mA @8MHz @3V
	Sleep	< 3.2 $\mu$ A	< 15 $\mu$ A	< 5.1 $\mu$ A
Radio	Tx	11.1mA	17.4mA	17.4mA
	Data Rate	0dBm / 2Mbps	0dBm / 250kbps	0dBm / 250kbps
	Rx	12.9mA	19.7mA	19.7mA
	Sensitivity	-85dBm / 2Mbps	-90dBm	-90dBm
	Sleep	22 $\mu$ A	20 $\mu$ A	20 $\mu$ A
Total	Active Tx	15.1mA (45.3mW)	25.4mA (76.2mW)	19.4mA (58.2mW)
	Active Rx	16.9mA (45.3mW)	27.7mA (83.1mW)	21.7mA (65.1mW)
	Sleep	< 25.2 $\mu$ A	< 35 $\mu$ A	< 25.1 $\mu$ A

\* The numbers are obtained from the datasheets of MicaZ and Tmote Sky [4], [12].

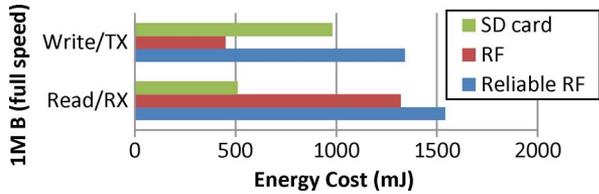


Fig. 4. Measured energy consumption comparison between SD card and RF.

TABLE IV  
RUNTIME REPROGRAMMING CODE SIZE WITH/WITHOUT ENIX

Application	Sensing & RF Transmit	Sensing & Log	RF Transmit & RF Receive	EcoNet Transmit	EcoNet Receive
Monolithic executable <sup>1</sup>	4825	2903	7646	8233	7770
Application on Enix <sup>2</sup>	474	673	514	1027	442
Application + TinyOS	9970	N/A	11286	N/A	N/A

unit: bytes

<sup>1</sup> Monolithic programs have no runtime support and handle drivers and other details by themselves.

<sup>2</sup> Code size of Enix is 8138 bytes.

and transmit them back to either the PC or cell phone. One prototype being developed tracks toddlers and warns parents wirelessly when the child wanders off or approaches a hazardous area.

## V. EVALUATION

### A. Power and Energy Consumption

Table III compares the power consumption of EcoSpire, MICAz, and Tmote. In active mode, even though the processor of EcoSpire consumes twice the power of Tmote at twice the clock rate, the overall EcoSpire system uses less total energy to transmit or receive RF packets due to higher level of integration and 8 $\times$  the data rate at 0 dBm RF power. The autoack and autoretransmit features also save significant amounts of power especially in a lossy environment.

Fig. 4 shows that reading from an SD card consumes much less energy than writing or RF receiving does. However, unreliable RF transmission consumes less than half the energy than SD writing does, while reliable RF transmission consumes more due to acknowledgment and retransmission.

### B. Runtime Support Using Enix

Table IV shows that the runtime support provided by Enix can significantly reduce the application code size. The first three applications implement basic sensing and data access while the

next two are EcoNet operations. The first one has a larger code size than the second one due the RF library. With the Enix OS on EcoSpire, the code size of all applications decreases significantly, because many functions are loaded on demand during runtime. Firmware upload time and power consumption during reprogramming can be reduced. Also, users can write concurrent programs supported by Enix, rather than having to serialize their code manually.

## VI. CONCLUSION

This letter presents the EcoSpire hardware and software kit for wireless sensor network applications. The node hardware achieves 8 $\times$  the data rate at the same power level as the latest design of the most popular motes. Moreover, our gateway enables integration of our ultra-compact platform with existing standards by not only bridging TCP/IP with heterogeneous radio standards but also supporting roaming. The tool chain, API library, runtime support with our Enix OS, and reference applications facilitate software development and generation of compact executables, thanks to support for concurrency and dynamic loading. Ongoing work includes integration with emulators and symbolic debugging.

## ACKNOWLEDGMENT

The authors would like to thank P.-N. Li for hardware manufacturing, and Y.-C. Li and Y.-L. Tsai for reference applications and documentation.

## REFERENCES

- [1] C. Park, J. Liu, and P. H. Chou, "Eco: An ultra-compact low-power wireless sensor node for real-time motion monitoring," in *Proc. 4th Int. Symp. Info. Process. Sens. Netw.*, Los Angeles, CA, 2005, pp. 398–403.
- [2] C.-Y. Ke, N.-Y. Ko, C.-H. Hsueh, C.-H. L. Lee, and P. H. Chou, "EcoPlex: Empowering compact wireless sensor platforms via roaming and interoperability support," in *Proc. 6th Annu. Int. Conf. Mobile Ubiq. Syst.*, Toronto, ON, Canada, 2009, pp. 1–9.
- [3] Mica2DOT [Online]. Available: [http://www.xbow.com/products/Product\\_pdf\\_files/Wireless\\_pdf/MICA2DOT\\_Datasheet.pdf](http://www.xbow.com/products/Product_pdf_files/Wireless_pdf/MICA2DOT_Datasheet.pdf)
- [4] MicaZ [Online]. Available: [http://www.xbow.com/Products/Product\\_pdf\\_files/Wireless\\_pdf/MICAZ\\_Datasheet.pdf](http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAZ_Datasheet.pdf)
- [5] MicaZ OEM [Online]. Available: [http://www.xbow.com/Products/Product\\_pdf\\_files/Wireless\\_pdf/MICAz\\_OEM\\_Edition\\_Datasheet.pdf](http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_OEM_Edition_Datasheet.pdf)
- [6] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling ultra-low power wireless research," in *Proc. 4th Int. Symp. Info. Process. Sens. Netw.*, Los Angeles, CA, 2005, pp. 364–369.
- [7] Tmote Mini [Online]. Available: [http://www.sentilla.com/pdf/eol/Tmote\\_Mini\\_Datasheet.pdf](http://www.sentilla.com/pdf/eol/Tmote_Mini_Datasheet.pdf)
- [8] P. Dutta, J. Taneja, J. Jeong, X. Jiang, and D. Culler, "A building block approach to sensornet systems," in *Proc. 6th Conf. Embed. Netw. Sens. Syst.*, Raleigh, NC, 2008, pp. 267–280.
- [9] S. Yamashita, T. Shimura, K. Aiki, K. Ara, Y. Ogata, I. Shimokawa, T. Tanaka, H. Kuriyama, K. Shimada, and K. Yano, "A 15  $\times$  15 mm, 1  $\mu$ A, reliable sensor-net module: Enabling application-specific nodes," in *Proc. 5th Int. Symp. Info. Process. Sens. Netw.*, Nashville, TN, 2006, pp. 383–390.
- [10] H. Cha, S. Choi, I. Jung, H. Kim, H. Shin, J. Yoo, and C. Yoon, "RETOS: Resilient, expandable, and threaded operating system for wireless sensor networks," in *Proc. 6th Int. Symp. Info. Process. Sens. Netw.*, Cambridge, MA, 2007, pp. 148–157.
- [11] 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 sensor networks," *Ambient Intelligence*, pp. 115–148, 2005.
- [12] Tmote Sky [Online]. Available: <http://www.eecs.harvard.edu/konrad/projects/shimmer/references/tmote-sky-datasheet.pdf>