TinyNode: A Comprehensive Platform for Wireless Sensor Network Applications

Henri Dubois-Ferrière
EPFL
1015 Lausanne
Switzerland
henri.dubois-ferriere@epfl.ch

Roger Meier
Shockfish SA
1015 Lausanne
Switzerland
roger@shockfish.com

Laurent Fabre
EPFL
1015 Lausanne
Switzerland
laurent.fabre@epfl.ch

Pierre Metrailler
Shockfish SA
1015 Lausanne
Switzerland
metrailler@shockfish.com

***Abstract*— We introduce the TinyNode platform for wireless sensor networks. Supporting both research and industrial deployments, the platform offers communication ranges that exceed current platforms by a factor of 3 to 5, while consuming similar energy. It comes with a rich, practical set of hardware extensions and full TinyOS support. We describe the design choices of the TinyNode, the accompanying hardware modules, and the MAC layer implementation.**

I. Introduction

Wireless sensor networks are emerging as an enabling technology for many applications which require as input quantities measured at multiple points in the physical world. However the requirements of different applications can vary widely in almost every aspect. In terms of hardware, a node for environmental or habitat monitoring [1] [2] will be placed outdoor and have multiple atmospheric sensors, whereas a node for embedded industrial applications (such as machine monitoring, [3]) may employ only a single vibration sensor but have stringent space requirements. A node for a research testbed is typically indoor, does not sense real phenomena, and requires a wired backchannel for control and measurements. In most deployments, a few basestation or clusterhead nodes require a longer haul, higher capacity connection such as WLAN or GSM. Energy requirements are also different, ranging from an ultra-low duty cycle application, which may survive on small batteries for several years, to higher duty cycle applications which require energy harvesting (e.g., of solar power) for autonomous long-term operation. Finally communication ranges may go from meters for highly dense deployments (e.g., microclimate monitoring [4]) to hundreds of meters for sparse networks covering large areas.

In this paper, we introduce the design of the TinyNode platform and its accompanying hardware extensions. The design philosophy of TinyNode is to place core components which are required for every application on a small PCB, and additional functionality on extension boards. The TinyNode core module is a versatile low-power sensor node, and comes with an array of extension hardware offering a wide set of connectivity, storage, energy, and interfacing options. It uses a low power transceiver which has energy characteristics comparable to those found on other sensor nodes ("motes"), but offers a signifi cantly larger range, and bit rates from 1.2kbps all the way up to 153kbps. The platform comes with full TinyOS support, including a complete radio stack, support for network reprogramming with Deluge, [5] and bridging software for GPRS/GSM data transfer.

II. TinyNode Core Module

The architecture of the TinyNode platform suite is organized around a core module (Fig. 1), and optional peripherals which can be selected based on application needs. Fig. 2 shows a block diagram of the core module, which contains the strict minimum common components required for operation: microcontroller, radio, flash, voltage regulator and supply monitor, and an expansion connector.

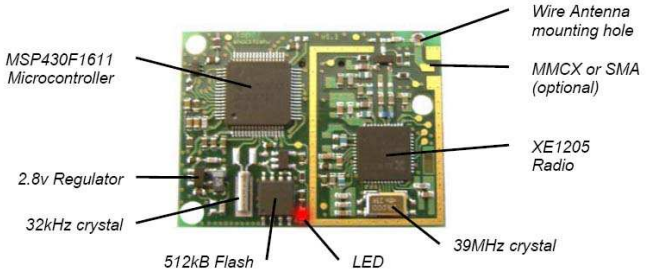

Fig. 1. TinyNode Core module (upper sides).

A. MSP430 Microcontroller

It features a MSP430F1611 ultra low power microcontroller that is fully supported by TinyOS and has the lowest power consumptions and fastest wake-up cycles available today. The digitally controlled oscillator (DCO) allows wake-up from low-power modes to active mode in less than 6μs and may operate up to 8MHz. Typically, the DCO will turn on from sleep mode in 300ns at room temperature. The MSP430F1611 has two built-in 16-bit timers, a fast 12-bit A/D converter, dual 12-bit D/A converter, one or two universal serial synchronous/asynchronous communication interfaces (USART), I2C, DMA, and 48 I/O pins. The same microcontroller is used on the Moteiv's Telos and on the Eyes platform (manufactured by Infi neon as the TDA5250 sensor node). We refer to [6] for a full comparison of the MSP430F1611 with competing

microcontrollers from Atmel, Motorola, and Microchip. The core module also has a 4Mbit flash chip that can be used for storing several firmware images or for logging data.

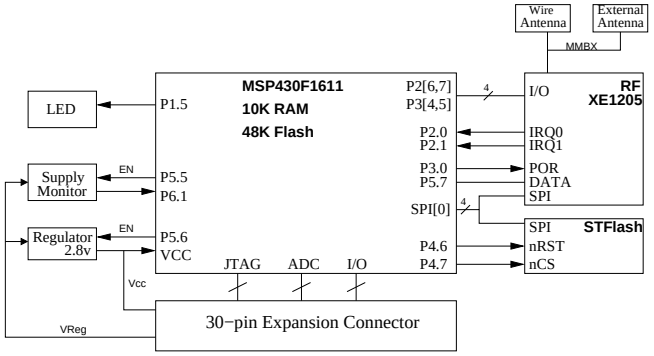

Fig. 2. TinyNode core module block diagram.

B. XE1205 Radio

Particular attention has been put to the choice of the radio transceiver. The XE1205 from Semtech (formerly XEMICS) is an integrated transceiver that can operate in the 433, 868 and 915MHz license-free ISM frequency bands. The current design of TinyNode supports European 868MHz operation; US versions are planned for 2006.

All major RF communication parameters are programmable and most of them can be dynamically set. The XE1205 offers the unique advantage of narrow-band and wide-band communication with the same hardware configuration. It can run data rates from 1.2kbit/s to 153kbit/s.

Compared to other transceivers in the market (including Chipcon, Nordic, RFM, Micrel, TI, Infineon), the XE1205 from Semtech offers the highest link budgets available today in the license free ISM bands (868 MHz in Europe or 900 MHz in US). With an output power of +15dBm and sensitivity of -116dBm at 4.8kbit/s, a link budget of 131dB can be achieved. This is around 22dB better than for Chipcon's CC1000 used on the Mica2 platform, which gives our platform around 4 longer range.

Table I shows the key transceiver characteristics for the CC1000, CC2420 (used on Telos and MicaZ), and XE1205 radio transceivers. The link budget is the sum of all signal gains and losses over the entire wireless path, and the receiver sensitivity is the signal level at which the decoded signal has a bit error rate (BER) of 0.1%. For comparison, the antenna gain is assumed to be unitary (0dBi) for all platforms and the outdoor range is calculated according to an isotropic path loss model [X] with a gain exponent of n=2.6 for open field propagation.

For low data rates, the RF frequency needs to be controlled carefully. The built-in frequency error indicator (FEI) of the XE1205 allows implementing an automatic frequency control loop (AFC) by software, avoiding the need for an expensive temperature compensated oscillator. An implementation of this loop under TinyOS is currently under work.

In addition to excellent range performance, the XE1205 has a modern zero-IF architecture that offers several advantages over traditional architectures that use one or more intermediate frequencies. Zero-IF means that the analog RF signal is directly converted to a digital baseband signal. Such architecture limits the number of required active blocks and avoids cumbersome external IF band pass or SAW filters for channel selection. Moreover, the baseband signal does not need to be DC balanced and can be NRZ coded. Compared to a Manchester encoded bit stream (typically used for Mica2 nodes), the effective data rate is double.

The internal 16 Byte FIFO buffer and the automatic pattern detector of the XE1205 reduce CPU load during time-critical transmit and receive loops. Using these features, we were able to run the radio at full 153kbit/s under TinyOS.

C. Energy consumption.

Energy consumption is a critical parameter of a sensor node. Table II shows the current consumption for Mica2, Telos, Eyes, and TinyNode. The MCU related consumptions of the TinyNode, Eyes, and Telos nodes are identical since they use the same chip. TinyNode has radio consumptions comparable to the Mica2, while offering significantly higher range and data rates. Telos also has comparable radio consumption, but the CC2420 offers a higher bit rate and faster radio wake-up. The tradeoff to this lower consumption is a reduced communication range.

III. XE1205 TinyOS Radio Stack

The port of TinyOS to the TinyNode platform consists essentially of low-level hardware adaptation code and a new radio driver and MAC layer for the XE1205 transceiver. The hardware adaptation phase made full use of the Hardware Abstraction Architecture (HAA) already developed at UC Berkeley and TU Berlin [7] with support for the MSP430 microcontroller. Our experience with this HAA has been very positive.

Unlike the core platform support, the radio stack had to be written from scratch, since the XE1205 transceiver has not been previously used in TinyOS-supported platform. We have designed and implemented a full radio stack around the XE1205 which includes CSMA, acknowledgement frames, low-power listening, and support for bit rates all the way up to 153kbps. The radio stack is relatively compact (206 bytes RAM and 6126 bytes ROM, including HPL and BusArbitration code).

The XE1205 interfaces to the microcontroller using SPI. It offers a bytewise read/write interface for sending and receiving data, and is configured with register operations over SPI. Support for full-speed operation at 153kbps would be difficult with a bare bytewise interface (such as that of the Chipcon CC1000), since every single byte (transmitted or received) must be handled in less than 50 μs. As a comparison, the TinyOS driver for the Mica2's CC1000 chip operates at 19.2 kbps, giving it up to 416μs to handle each byte.

Fortunately, the XE1205 includes some functionality which helps to offload the microcontroller. In particular, it offers a 16-byte FIFO buffer for sending and receiving packets, and a hardware preamble detector which generates an interrupt as

Platform	Mica2		Telos Sky	Eyes		TinyNode	
Transceiver	CC1000		CC2420	TDA5250		XE1205	
Frequency	869 Mhz		2.4 Ghz	869 Mhz		869 Mhz	
Max. Tx Power	5dBm		0 dBm	9dBm		15dBm	
Data Rate	76.8 kbps	4.8 kbps	250 kbps	64 kbps	4.8 kbps	76.8 kbps	4.8 kbps
Sensitivity	-98 dBm	-104 dBm	-94 dBm	-96 dBm	-110 dBm	-106 dBm	-116 dBm
Link Budget	103 dB	109 dB	94 dB	105 dB	119 dB	121 dB	131 dB
Range Outdoor[1]	160m	300m	80m	200m	600m	600m	1800m

TABLE I

COMPARISON OF RADIO TRANSCEIVER CHARACTERISTICS.

	Mica2	Telos Sky	Eyes	TinyNode	
Min Voltage	2.7	1.8	2.1	2.4	V
Max Voltage	3.3	3.6	3.6	3.6	V
MCU sleep with RTC on (LPM3)	19	5.1	5.1	5.1	μA
MCU active	8	1.8	1.8	1.8	mA
MCU active, Radio RX	15.1	21.8	10.8	15.8	mA
MCU active, Radio TX at +0dBm (1mW)	25.4	19.5	13.7	25	mA
MCU active, Flash Read	9.4	4.1	5	5	mA
MCU active, Flash Write	21.6	15.1	16	16	mA
MCU wake-up latency	180	6	6	6	μs
Radio wake-up latency	1800	580	2200	1500	μs

TABLE II

CURRENT CONSUMPTION AND WAKE-UP TIMES.

soon as a configurable preamble (of length 8 to 32 bits) is received. While the FIFO buffer avoids having to read (write) every byte as it arrives (transmits), a latency below 50 μs is still necessary during packet reception of long packets each time the FIFO reaches 16 bytes, and during transmission each time the FIFO's becomes empty. At this point, the driver must respond rapidly enough to read (write) 16 bytes from (into) the FIFO, otherwise an incoming byte will be lost, or the outgoing bitstream will contain a gap and lose synchronisation.

To evaluate software overhead, we measured processing and switching by sending a continuous packet stream, with initial backoffs disabled, and computing the total channel utilization. While this is not a realistic application profile, it allows us to evaluate if there is any inefficiency in the packet-processing and switching times. Results show that the driver is fast: total channel utilization when sending a continuous packet stream, is 68.8% at 152kbps, 80.2% at 76kbps and 94.7% at 19.2kbps. In comparison, the Mica2 stack running at 19.2 kbps has approximately 85% channel utilization in the same conditions. For TinyNode, utilization decreases with bit rate, because the per-packet overhead has a large constant component which is independent of bit rate.

A. Low Power Listening Implementation

For duty cycling, the XE1205 radio stack implements a technique known as low power listening (or preamble sampling) [8] [9]. Low power listening achieves a low duty cycle by having nodes periodically awaken for short periods and listen on the radio. If a node detects an ongoing packet preamble transmission, it remains awake to receive the packet; otherwise it returns to sleep until the next wakeup time. A transmitter sends a packet with a preamble of length sufficient to cover the receiver sleep period. This technique has the advantage of being simple and robust and does not require node synchronization or any other form of coordination between neighboring nodes.

The implementation takes advantage of the XE1205 pattern detector: when a node wakes up, it programs the pattern detector with a two-byte pattern 1010101.., corresponding to the preamble sequence. If these bytes are received, the radio will signal an interrupt, and the node now knows that (with high probability) a packet preamble is ongoing. It then reprograms the preamble detector with a 3-byte start-of-frame sequence, and awaits a second interrupt signalling the start of packet reception.

For real performance, it is interesting to examine a typical low power listening (LPL) mode with 1% duty cycle, meaning that the receiver is active during 1% of the time for listening. At 152kbit/s, we obtain a minimum listen period of 1.9ms (including radio start-up time and RSSI measurement), which means the listening period is 190ms for a 1% activity. In comparison, due to higher start-up times and lower data rates, a Mica2 node at 19.2kbit/s has 8ms of listening time and a listening interval of approximately 1 second. In comparison, this represents a fivefold improvement in latency (or equivalently, throughput) over Mica2, with comparable battery lifetime and range. Conversely, if an application can tolerate a 1-second per-hop latency but requires minimizing energy consumption, the TinyNode can run at 0.2% duty cycle and consume almost an order of magnitude less than Mica2, whilst offering the same delay as the Mica2 at 1% duty cycle. In this case, a theoretical lifetime of over 6 years can be achieved with 2 x AA alkaline cells. Note that the relative improvement is smaller than the ratio of bit rates, because both listen times include a radio wake-up time. While the XE1205 wake-up time is shorter than the CC1000 wakeup time, its relative duration when counted in byte times at 153kbps is higher than for the CC1000. Table III summarizes these numbers. Note that

due to self-discharge and degradation, another type of battery technology needs to be used to obtain such lifetimes (such as Lithium Thionyl Chloride).

B. Radio Range

The hardware characteristics of the XE1205 transceiver shown in Table I indicate a theoretical range of up to 1800m at low bit rates. While link budget, sensitivity, and theoretical ranges allow for high-level comparison between different transceivers, it is necessary to validate them with empirical measurements in order to ascertain that entire system performance is in line with expectations.

We ran a simple experiment in order to show the limit of range that can be achieved using TinyNode at 4.8kbit/s and maximum transmit power. We placed a transmitter on the balcony of a 4-floor building and the receiver on the roof of a car. Both had good $1/4$ wave monopole antennas with approximately 0dBi of gain. The sender sends a short 6 byte frame and the receiver blinks its LED each time it receives a correct frame.

We then drove a circuit with the car while observing the connection. The trace of this experiment is shown in Fig. 3. During most of the circuit, we were not in line of sight of the transceiver. The only line-of-sight segment is a 200m portion of the road in the immediate vicinity of the transmitter. The area of this experiment is a hilly (elevations are shown on the map in italicized font) urban and suburban area. The solid line indicates a reliable link from our node in a moving car to the receiver on the building, and the dotted line indicates areas of unreliable reception (2). The unreliable portion corresponds to approximately 10% of the circuit. We were able to communicate up to 2.3 km (North segment near 'Crochy') without direct line-of-sight.

This experiment confirms (and exceeds) the theoretical outdoor range of Table I. This outdoor range is higher than that of widespread platforms (such as Mica2 or Telos) by a factor of 4 to 8, both in comparison with the theoretical numbers of Table I, and with radio ranges found in previous experimental studies [10] [11] [6]. With this radio range, TinyNode enables network spans that were previously not possible. Even assuming a highly conservative average range estimate 250m, a sizeable urban area can be covered without requiring thousands of nodes. For example, a city such as San Francisco (approximately 8km by 8km) could be covered with approximately 1000 nodes, whereas with a radio range of 80m this would require over 10000 nodes.

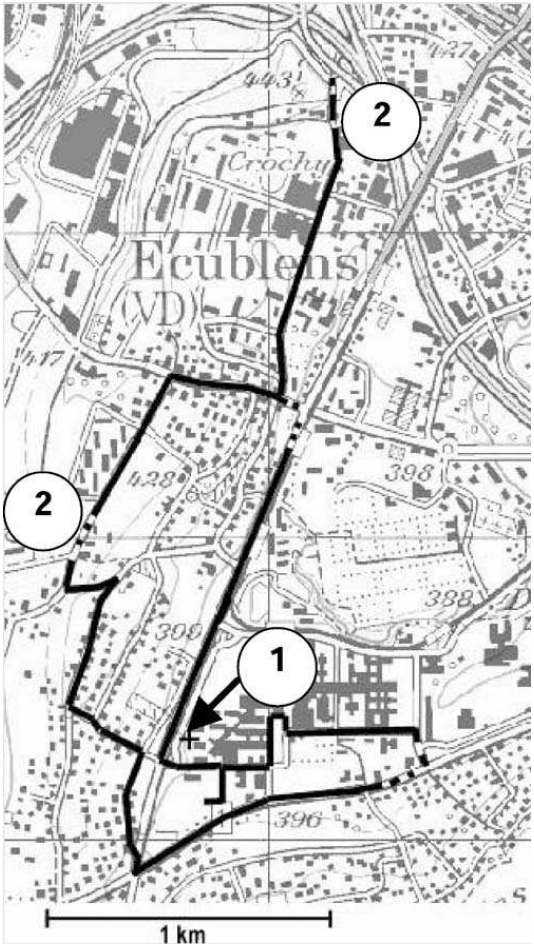

Fig. 3. Range Trace. (1): Transmitter position, on a 4th floor balcony of a building. The solid line indicates a reliable link from a node in a moving car to the receiver in the car, and the dotted line indicates areas of unreliable reception (2). All parts of the route were non line-of-sight, except for a 200m portion of the road in the immediate vicinity of the transmitter.

IV. Extension Boards

A. Standard Extension Board

The Standard Extension Board (SEB) is designed as a low-cost extension to the TinyNode and provides the necessary functionality for development purposes, simple deployments and hardware prototyping. It is a strict subset of the MamaBoard; in other words all of the hardware and functionality of the SEB is also found in the MamaBoard.

The SEB can be powered either from battery or from an external supply through a Jack connector. For programming and debugging, it includes a JTAG interface and a RS-232 with full BSL support. The RS-232 drivers are powered directly from the serial line, avoiding drawing current from the battery. The board includes footprints for two optional sensors: a relative humidity and temperature sensor (Sensirion SHT11), and a photodiode light sensor (Infineon BPW345-P1602). Elements for basic interaction with a running system are 3 light-emitting diodes (LEDs) and 4 Jumpers that can be read on the microcontroller's digital inputs.

The SEB also offers a pad field that can be used to quickly connect custom sensors or actuators. The pad field has a row of unused pins coming from the microcontroller on the one side and a row of pads that are connected to a standard flat cable connector on the other side. In the middle, there are unused pin-through-hole pads that can be used to solder simple glue electronics.

B. MamaBoard

In addition to the functionality of the SEB, the MamaBoard offers rich connectivity and storage options, allowing to bridge a wireless sensor network to wired ethernet (LAN), WLAN, or

	Mica2 at 1%	TinyNode at 1%	TinyNode at 0.2%
Bit Rate	19.2 kbps	152 kbps	152kbps
Listen Time	8 ms	1.9 ms	1.9
Listen Period (Max. Latency)	1085 ms	190 ms	950 ms
Max throughput	0.89 pkts/sec	5.5 pkts/sec	1.05 pkts/sec
Average Power Consumption	509μW	489μW	104 μW
Lifetime[2] for 2 x AA alkaline cells, 2000mAh	1.3 years	1.4 years	6.6 years

TABLE III

CURRENT CONSUMPTION AND WAKE-UP TIMES.

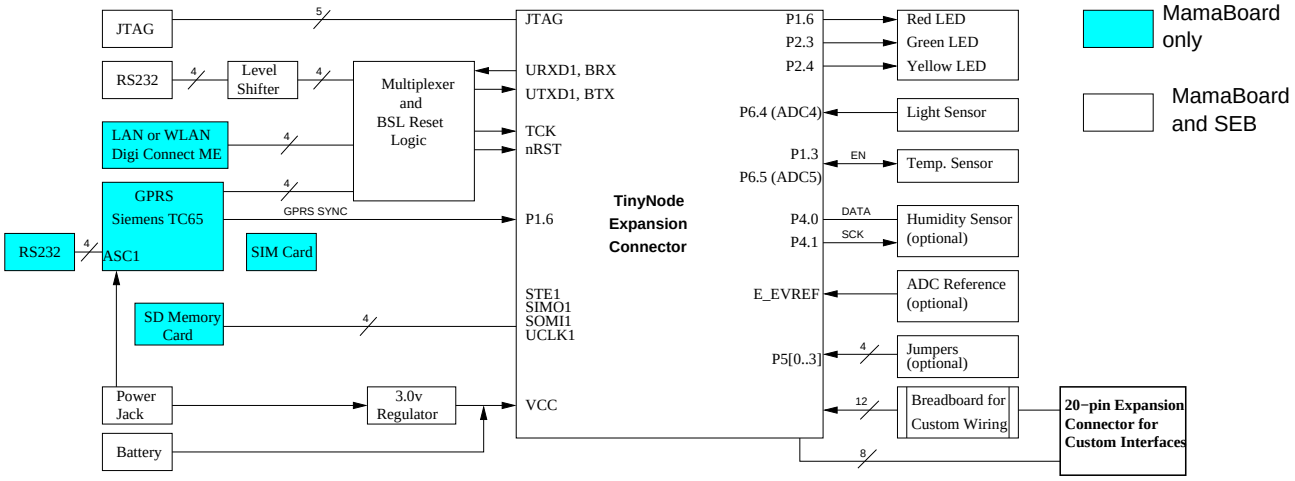

Fig. 4. SEB/MamaBoard block diagram.

cellular GPRS. These three connectivity types are achieved by plugging optional external modules onto the board. For robust and low-cost mass storage, the board includes a SD memory card slot which can be accessed both by the TinyNode and by the GPRS module.

1) LAN and WLAN connectivity: LAN connectivity is achieved by plugging a Digi Connect ME [12] and WLAN connectivity with a Digi Connect Wi-ME. Both modules are IP-capable devices that offer transparent serial port relaying over TCP/IP, allowing a host PC to "mount" the serial port over the network and interact with the remote TinyNode as if it were locally connected to the PC's serial interface. Using the wired ethernet option, a testbed of TinyNodes with wired backchannel can be set up with little effort and at low cost (the Digi Connect ME can be purchased for approximately $50). We have recently installed a 50-node testbed at EPFL using the Digi Connect ME as a wired backchannel. The Digi Connect Wi-ME offers the same functionality as the ME, but over WLAN.

2) GPRS/GSM Connectivity: In addition to Ethernet and WLAN, the third connectivity option provided by the MamaBoard is GPRS. GPRS is a packet-switched mobile data service available in GSM cellular networks. With GSM/GPRS connectivity, we aim for the following requirements: (i) sending measurements received on the MamaBoard-hosted TinyNode to a central server over the Internet, (ii) remotely controlling the MamaBoard by SMS, and (iii) performing software updates of all nodes attached to the wireless sensor network reachable from the MamaBoard.

Besides a very high latency, IP over GPRS faces two issues. First, the quality of the connection varies greatly and the connection itself is often lost. Second, most mobile operators place their GPRS network behind a fi rewall and assign private (non-visible) addresses to a GPRS node, thus barring a central server from initiating a connection from a central server to the GSM/GPRS module. It is however possible to send SMS data messages to the module, since it is a full GSM device.

For the application that resides on the GPRS module we have designed two modes of operation that address these issues and achieve the above-mentioned requirements: client mode, and bridge mode.

Client mode. The GPRS module (a) listens to packets sent on the serial port by the MamaBoard, (b) translates the packets into a suitable format required by the industrial application on the central server, typically a flat fi le or a XML/SOAP message, and (c) sends them over the internet to the central HTTPS or FTP server. If the GPRS link fails, the application will buffer the packets. In this mode, the central server has no knowledge of the underlying protocol, for example the TinyOS message format.

Bridge mode. Upon receiving a special SMS, the GPRS module establishes a connection to a bridge service that resides on the central server. This bridge service listens on two ports, one for the GPRS module and one for the local forwarding service. Such a local forwarding service could for example be the TinyOS SerialForwarder. When both ports have accepted a connection, the bridge acts as a stub and forwards the packets between the two endpoints. The server application, using the local forwarding service, is transparently connected to the

GPRS module. This mode is particularly useful for performing software updates on the wireless sensor node by typically using the Deluge suite.
Besides these two modes of operation, the GPRS module generates interrupts for incoming SMS and handles over-the-air provisioning (OTAP) that enables software updates of the GRPS module itself from a remote location.
We selected the Siemens TC65 Wireless Module for its Java 2 Micro Edition (J2ME) built-in support and its versatile peripheral interfaces (serial, I2C, SPI). Running Java on the module is clearly an advantage because the TinyOS toolchain reference is completely based on this language and the GPRS module has to implement some of the functionality contained in the toolchain. From the hardware perspective the TC65 features a quad-band GSM technology for worldwide use, an ARM7 processor, 1.7MB of Flash memory and 400kB of RAM. This footprint leaves a reasonable space for buffering packets when the GPRS connection is lost.
The complete application that resides on the GPRS module is implemented in Java on the TC65. As the J2ME is a subset of the Standard Edition it was not possible to embed the TinyOS Java communication stack as-is, but we were still able to reuse a fairly large amount of existing code, and to write a new MIG target in order to generate TinyOS message classes for the J2ME.
Siemens is currently the only vendor offering such a rich but lightweight platform. Other manufacturers, notably Sagem, are planning to add Java support to their existing solutions and launch similar products. Siemens, as well as other vendors, offers cheaper GPRS modules such as the MC55 that features an IP stack accessible by AT commands from the serial port. These modules do not embed custom applications and are therefore unable to meet our requirements
3) SD Memory Card: MamaBoard also provides a Secure Digital (SD) card slot for mass storage. SD is a flash memory card format that has become widely used in portable devices. Typical capacities today are 128, 256 and 512 megabytes, 1, 2 and 4 gigabytes. Interfacing with the TinyNode is straightforward, since all SD memory cards are required to support the older SPI/MMC serial mode that is compatible with the MSP430 SPI ports. The SD card can be used as a robust and low-cost mass storage device for data logging applications or as a temporary buffer, in case the connectivity to the MamaBoard is lost. It can be accessed both by the TinyNode and the GPRS module.

V. Solar supply

We have also designed a solar energy supply board for long-term outdoor operation. This system operates with a small solar panel, a primary energy buffer, and a secondary energy buffer. A block diagram is shown in Fig. 5. The overall architecture is similar to that of Prometheus [13]: a primary buffer is designed to collect energy from the solar panel, powering the node whenever possible, and a secondary buffer, with higher capacity, is used for extended operation when solar energy is not sufficient to fill the primary buffer.
Our first goal in the design of this system was *flexibility* of hardware configurations, in order to meet widely varying energy budgets of different applications. At one extreme, a low rate environmental monitoring application may require nodes to sample sensors and send data at intervals of one minute or more. This can be achieved with a duty cycle of 1% or lower. At the other extreme, one can consider a system where a node's radio must be in continuous operation, because of a high traffic rate, or to meet stringent latency requirements. Of course, average duty cycle is only one characterization of energy consumption, and a detailed application analysis should include the individual power consumption and duty cycles of each sensor, as well as the variance (or distribution) of idle periods when workloads are not strictly periodic. But the key point remains: there is no set of solar panel, primary, and secondary buffer that is both cost- and performance-optimal for an energy consumption range of 2 or more orders of magnitude. At the same time, it is not practical to design a different energy board for each application. Our approach is therefore to design a single PCB board which contains all the necessary circuitry and interfacing, whilst allowing a large degree of energy flexibility by selecting different options for the solar panel, primary, and secondary buffer.
Our second goal is to offer complete *control and monitoring* interfaces to the attached node that. By allowing a node to control energy flows and by exposing full monitoring of the energy state to the node, it is possible to manage the energy subsystem in software. This gives the flexibility to adapt the power management strategy to different choices of components by simply reprogramming a node with the appropriate algorithm and parameters. Specifically, the node can measure the following quantities: primary buffer voltage U_{prim}, secondary buffer voltage U_{sec}, external power supply voltage U_{ext}, the solar panel current I_{solar}, the charge current of the primary buffer I_{charge} (0 when not charging), and the overall current consumption of the TinyNode and any additional sensors I_{supply}. We use two ADC ports on the MSP430 microcontroller for these measures: one for voltages, and one for currents. We use a current monitor (Zetex ZXCT1010) to measure the currents using the MSP430's ADC ports. The input on either ADC port is selected using two multiplexers which are controlled with two digital outputs from the MSP430.

A. Choice of components

Duty Cycle	2x 1F supercap. 1.9 mWh 1.6V-5.5V	2x 22F supercap. 42.3 mWh 1.6V-5.5V	150mAh Ni-Mh bat. 720 mWh 5.3V-4.4V
1%	49 minutes	18.3 hours	12.8 days
10%	4.9 minutes	1.8 hours	31 hours
100%	29 seconds	11 minutes	3 hours

TABLE IV
COMPARISON OF PRIMARY AND SECONDARY ENERGY BUFFER OPTIONS.

The key system components are the solar panel, primary buffer, and secondary buffer. Our criterion for the solar panel was that it should generate sufficient power to charge the primary buffer in less than a day. This should hold even in

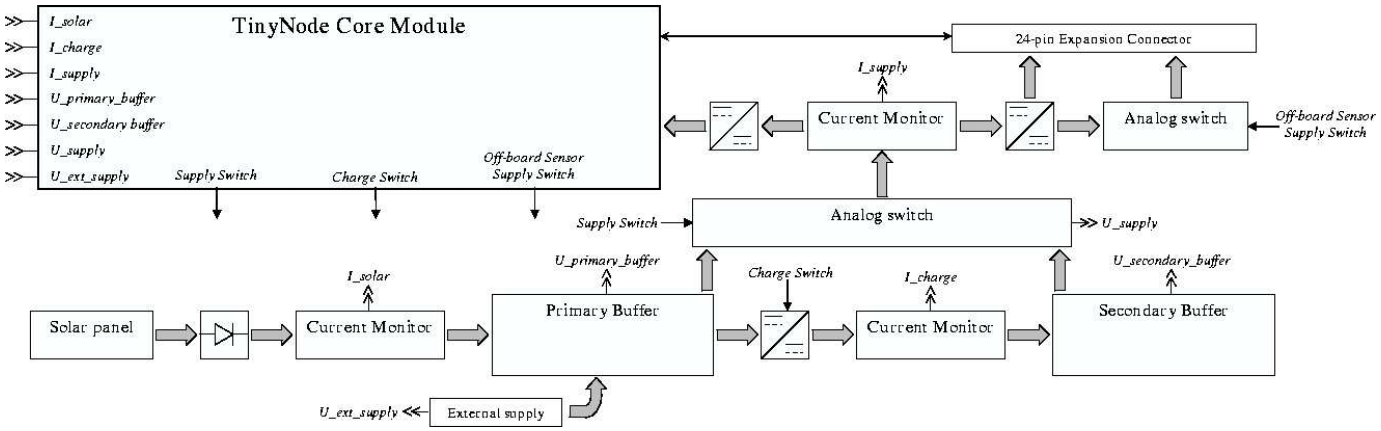

Fig. 5. Solar supply block diagram.

poor luminosity conditions, in order for the energy subsystem to work even in shorter winter days, including those without sunshine. We settled on a 140x40mm solar panel with has a nominal power output of 300mW in direct sunlight. Other solar panel options for different power requirements include panels of different dimensions (as long as they provide a 5-7V voltage) or wiring (a small number of) panels in parallel.

The role of the secondary buffer is to offer an energy backup source that can allow the node to operate for periods when the solar panel is not providing sufficient power to drive the primary buffer. The length of such black-out periods depends essentially on weather factors and on the geographic location of a deployment. We are currently using a Li-Ion battery [14] from Leclanché which has a capacity of 2000mAh under 3.7 Volts. This choice is motivated by a future deployment on an Alpine glacier. In this deployment, the nodes' radio duty cycle will be on the order of 1%, but energy requirements are dominated by a multi-sensor which will be continuously drawing 8mA current. This system will therefore run for 250 hours on the battery, allowing it to survive until a maintenance visit if a solar panel becomes covered in ice or snow.

The final component to choose is the primary buffer. We selected three options covering a large capacity range. These are summarized in Table IV. Each option is matched to different requirements. The first option has lowest capacity. It uses two 1F supercapacitators [15] which are wired in series in order to obtain an operating voltage of up to 5.5V. This gives a total capacitance of 0.5F. It has lowest cost and is matched to a system where good solar exposure can be expected on a near-daily basis, allowing the node to charge the secondary buffer daily. The second option uses two 22F supercapacitators, giving a capacitance of 11F. This solution is the most expensive, but allows a node to operate for much longer without drawing from the primary capacity. Finally, we can also place a 150mAh Ni-Mh rechargeable battery as primary buffer. Its energy capacity is superior to the 22F supercapacitators by a factor of 17, and it is also cheaper. Its disadvantage is that it tolerates a finite number of approximately 1000 charge cycles, unlike supercapacitators which have unlimited charge cycles.

Table IV summarizes the three options for the primary energy buffer, and gives theoretical node primary buffer duration for three different duty cycles.

B. Algorithm and Measurements

We implemented a first driver in TinyOS based which follows the charging strategy of Prometheus [13]. We tested the performance of our board with this algorithm, using three nodes: Node A had a 22F supercap primary buffer, Node B had a 150 mAh Ni-mH battery, and Node C had a 1F supercap. The algorithm constants were different than in the Prometheus implementation and were computed individually for each configuration; we omit the details for lack of space. The three nodes were placed on a rooftop at EPFL for a one-week period in November 2005. Each node had a 10% radio duty cycle, representing a current draw of approximately 2mAh. Nodes measured and transmitted $U_{sec}, U_{prim}, I_{solar}, I_{charge}$, and I_{supply} every 10 seconds to a basestation in a neighboring office. All systems performed satisfactorily.

We focus now on the behavior of nodes A and B. We omit Node C, which had a more monotonic and predictable behavior with very rapid charge/discharge cycles of its smaller primary buffer. Figure 7 shows the data collected in a three day period. Day one was cloudy without any sunshine, and days two and three were clear and sunny (though with a fairly low winter sun). This is clearly observable in the bottom plot of solar panel output current. We can observe that for node A, the Li-Ion accumulator had clearly charge increases on days two and three, but not (or barely perceptible) on day one, whereas node B did increase the battery charge even on day one. This is due to the fact that node B's primary buffer is an order of magnitude larger than node A, and was still quite full going into day two. A small amount of light was sufficient to increase it to to 5.3V, at which point the driver initiated a charge cycle. Note that on day three, the voltage of node B's Li-Ion battery does not increase because it is fully charged at 4.2V. We now turn to the primary buffers. On node A, we can clearly see the daytime periodic charge cycles. Once the supercapacitators reached 5.4V, the charge switch opened into the Li-Ion battery until the supercapacitator ran down to 4V. These cycles are significantly slower on day one. We can see that on days two and three, node A does not quite make it through the night on the supercapacitator, and must switch to the secondary buffer

for the last four hours of the night. This is not due to an insuffi cient capacity, but to simply because the capacitator was in the middle of a charge cycle when nightfall came. Without a control loop, the voltage of the capacitator will be randomly distributed between it's upper and lower operating points. Such a control loop should decrease the length and depth of the charge cycles toward the end of the day so as to start the night with a full primary buffer; we will implement one in the next iteration of the driver. Finally, we observe that node B never needed to use the secondary buffer in the entire period. In fact, with our 10% duty cycle node, the overall energy balance is clearly positive, since we can see that the secondary buffer is fully charged from the morning of day three (a Zehner diode prevents it from charging beyond 5.5V).

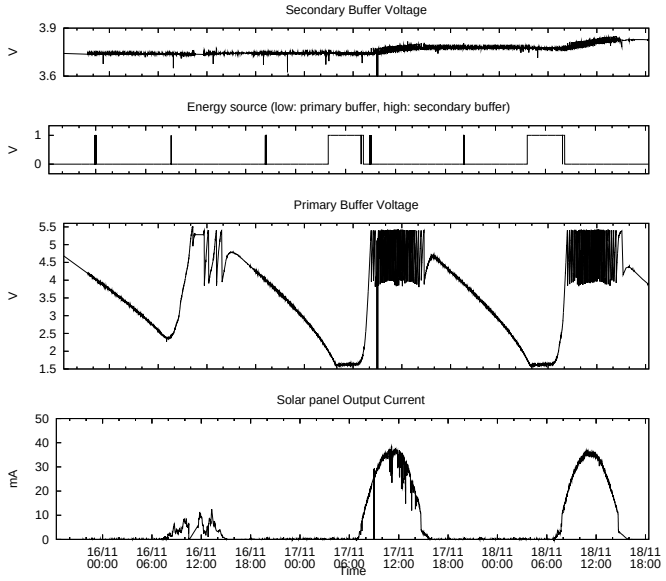

Fig. 6. 72-hour energy trace for node A (2 x 22F Supercapacitator).

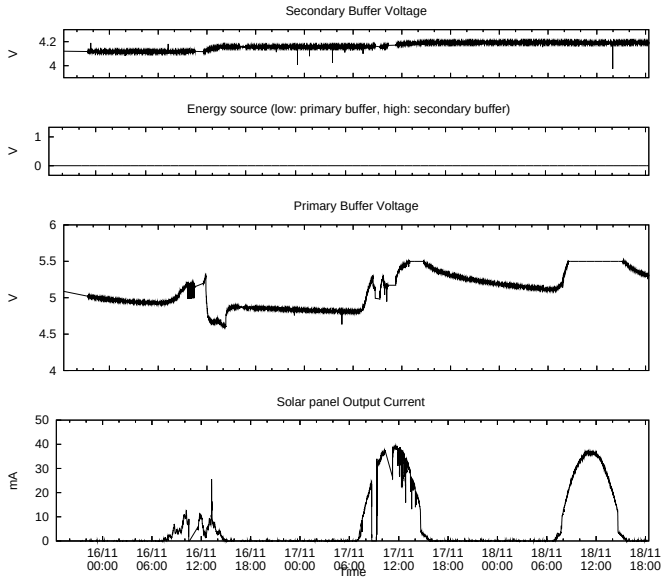

Fig. 7. 72-hour energy traces for node B (150 mAh Ni-mH battery).

VI. Conclusion

We have presented TinyNode, a new wireless sensor networking platform which comes with a rich set of hardware extensions for backhaul connectivity (ethernet, WLAN, and GPRS), mass storage, solar energy harvesting, and custom interfacing. The core module itself is a small, low power device, and uses a radio transceiver with range characteristics far beyond existing platforms. The node runs TinyOS with a complete radio stack and support for key subsystems such as network reprogramming. Several deployment projects with TinyNodes are under development in the areas of environmental monitoring, precision agriculture, and parking management.

References

[1] Alberto Cerpa, Jeremy Elson, Deborah Estrin, Lewis Girod, Michael Hamilton, and Jerry Zhao, "Habitat monitoring: Application driver for wireless communications technology," in *2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, Costa Rica, 2001.

[2] Robert Szewczyk, Alan Mainwaring, Joseph Polastre, and David Culler, "An analysis of a large scale habitat monitoring application," in *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Baltimore, November 2004.

[3] Robert Adler, Phil Buonadonna, Jasmeet Chhabra, Mick Flanigan, Lakshman Krishnamurthy, Nandakishore Kushalnagar, Lama Nachman, and Mark Yarvis, "Design and deployment of industrial sensor networks: Experiences from the north sea and a semiconductor plant," in *Proceedings of ACM Sensys*, San Diego, USA, November 2005.

[4] Gilman Tolle, Joseph Polastre, Robert Szewczyk, Neil Turner, Kevin Tu, Phil Buonadonna, Stephen Burgess, David Gay, Wei Hong, Todd Dawson, and David Culler, "A macroscope in the redwoods," in *Proceedings of ACM Sensys*, San Diego, USA, November 2005.

[5] Jonathan W. Hui and David Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, New York, NY, USA, 2004, pp. 81–94, ACM Press.

[6] Joseph Polastre, Robert Szewczyk, and David Culler, "Telos: Enabling ultra-low power wireless research," in *The Fourth International Conference on Information Processing in Sensor Networks: Special track on Platform Tools and Design Methods for Network Embedded Sensors (IPSN/SPOTS)*, Los Angeles, California, Apr. 2005.

[7] V.Handziski, J. Polastre, J.-H. Hauer, C. Sharp, A. Wolisz, and D. Culler, "Flexible hardware abstraction for wireless sensor networks," in *Proc. of 2nd European Workshop on Wireless Sensor Networks (EWSN 2005)*, Istanbul, Turkey, Feb. 2005.

[8] Joe Polastre, Jason Hill, and David Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of ACM Sensys*, Los Angeles, USA, April 2003.

[9] A. El-Hoiydi, J. Decotignie, and J. Hernandez, "Low power mac protocols for infrastructure wireless sensor networks," 2003.

[10] Jerry Zhao and Ramesh Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proceedings of ACM Sensys*, Los Angeles, USA, April 2003.

[11] Alberto Cerpa, Naim Busek, and Deborah Estrin, "Scale: A tool for simple connectivity assessment in lossy environments," in *CENS Technical Report 0021*, 2003.

[12] Digi International, "Digi connect ME and digi connect Wi-ME," http://www.digi.com.

[13] Xiaofan Jiang, Joseph Polastre, and David E. Culler, "Perpetual environmentally powered sensor networks.," in *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks, IPSN 2005, April 25-27, 2005, UCLA, Los Angeles, California, USA*. 2005, pp. 463–468, IEEE.

[14] Leclanché, "Lgc-1865," http://www.leclanche.ch/fr/produits/accumulateurs/li ion.php.

[15] Cooper Industries, "Aerogel supercapacitators, b series," http://www.cooperet.com/library/products/PS-5102 B Series.pdf.