TelosW: Enabling Ultra-Low Power Wake-On Sensor Network

Gang Lu Debraj De Mingsen Xu Wen-Zhan Song
Sensorweb Research Laboratory
Washington State University
Email:{gang_lu, debraj_de, mingsen_xu, songwz}@wsu.edu
Jiannong Cao
Department of Computing
Hong Kong Polytechnic University
Email: csjcao@comp.polyu.edu.hk

***Abstract*—Sensor networks are typically sensor or radio event driven. Exploiting this property we propose a novel wake-on sensor network design. In this context we have designed a new sensor platform called $TelosW$. The wake-on sensing capability of $TelosW$ lets designated sensors wake up the microcontroller(MCU) only on occurrence of some event with preconfigurable threshold. $TelosW$ also includes the CC1101 [3] Wake-On Radio (WOR) hardware that performs low power listening without intervention of MCU. These all lead to a completely event driven wake-on sensor network that reduces energy consumption considerably. $TelosW$ is also equipped with an on-board energy meter that can precisely measure in-situ energy consumption. Using the energy meter it is possible to get the insight of energy states of nodes in a network at any time. This makes it possible to practically analyze energy-efficient protocols. The experiments show that the energy consumption has been significantly reduced comparing to same application without wake-on design.** [1]

***Index Terms*—TelosW, energy meter, wake-on radio, wake-on sensor.**

I. Introduction

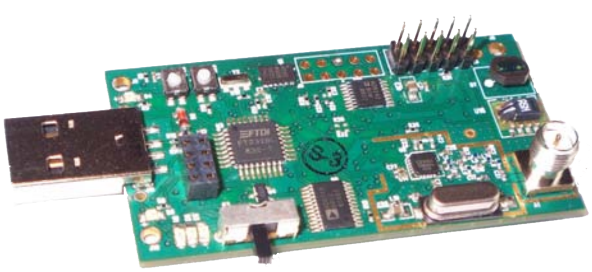

Fig. 1. TelosW platform

Significant advancement in wireless communication and microelectronic technologies have revealed the great potential of Wireless Sensor Networks (WSN). Wireless Sensor Networks have been used for variety of applications such as scientific exploration [16], infrastructure protection [18], surveillance [10], assisted living [17] etc. Despite its research, development and deployment through years, there are a number of open issues towards achieving its full potential. Energy efficiency is a key goal of wireless sensor networks design as it decides its efficiency, lifetime, performance etc. Therefore it has been a major goal to minimize the energy consumption of individual node, as well as the network as a whole for collective operations.

A typical nature of sensor networks is that the most of the activities are event driven. But in current sensor networks, the sensing and communication components are even powered on during significant portion of idle time. This obviously leads to a large amount of undesirable wastage of energy. These all have motivated us to utilize the event driven properties of sensor networks into sensing and communication, that leads to significant energy savings. In this perspective we propose a novel wake-on wireless sensor network, enabled with a hardware platform which supports wake-on capability in sensing and communication.

Proper knowledge of in-situ energy consumption is a crucial factor towards informed decision making in wireless sensor networks. Towards the novel effort to measure detailed energy consumption on sensor node for free (at almost no cost of extra energy), the work by Dutta et al. [7] proposes an energy meter hardware design. This free energy metering can be utilized at scale in a distributed environment of sensor network. This will give the valuable insight into the states of the nodes in a network of any size. In this perspective we have used the energy meter design to enrich the capability of our newly designed sensor platform.

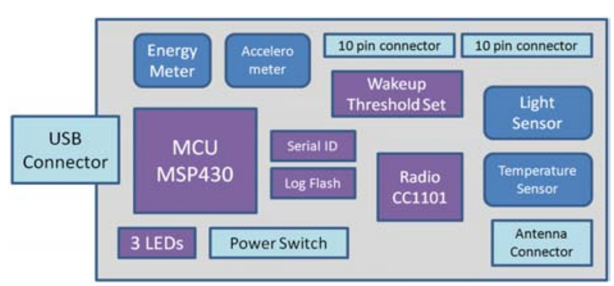

Fig. 2. Components of TelosW

[1]This work is partially supported by NSF-CNS-0914371, NSF-CNS-0953067 and Hong Kong ITF Grant

The $TelosW$ sensor platform (shown in Figure 1 and Figure 2) enables the design of an ultra-low power wake-on sensor network. $TelosW$ has a wake-on hardware feature that lets the on-board sensors wake up the MCU only on the occurrence of configurable event levels. This kind of wake-on operation lets the MCU sleep entirely during idle time. $TelosW$ also includes the wake-on radio hardware that enables event-driven wireless communication. These all event driven sensing and communication operations lead to savings in energy consumption and lower load for MCU. Through experiments described later, it is observed that wake-on sensor design also ensures that no event is missed from detection. But nodes without wake-on sensor have to spend more energy in order to maintain event detection accuracy. Using all these properties of $TelosW$, we have designed a wake-on sensor network, where the sensing and communication are entirely event driven. Application of this wake-on design in hardware platform and completely event driven operations in distributed environment promise a major shift in the way sensor network operates. In future sensor networks all the components will be in ultra-low power sleep state. Only desired and configurable events will trigger relevant operations in the related region of the network. Then the activated components will again go to sleep state. These will result in achievement of a true ultra-low power sensor network.

In our developed platform $TelosW$, we have included the Wake-On Radio (WOR) CC1101 [3]. This enables the even driven wake-on communication. Besides the WOR capability, the radio also supports multiple data rate, multiple power levels and multiple operating channels. Multiple data rate with multiple power level provides a variety of transmission range. These further enrich the capability of $TelosW$ enabled wake-on sensor network for a large variety of applications and deployment scenarios.

Fig. 3. Lab setting for experiments

The main contributions of this paper are as follows:

- We have designed a novel sensor platform $TelosW$ with sensor wake-on hardware that enables configurable event driven sensing capability.
- We have utilized the Wake-On Radio (WOR) hardware to enable wake-on communication capability on $TelosW$.
- We have used an on-board energy meter for validating the energy efficiency of $TelosW$ operations. It is also used for analyzing the in-situ distribution of energy consumption precisely across nodes in a network.

The rest of this paper is organized as follows. Section II presents the design of $TelosW$ hardware platform in detail. Section III profiles energy saving on $TelosW$ with wake-on sensor and with wake-on radio. Section IV shows the utilization of energy meter for analyzing network-wide distribution of energy consumption. Section V presents the test result of transmission range with different data rate. Section VI explores the background and related work. Finally we conclude in section VII.

II. HARDWARE: TELOSW PLATFORM

We have developed $TelosW$, which is a sensor device with wake-on capability. In this section we will describe the design and key features of $TelosW$ hardware.

A. Overview

The most important feature of $TelosW$ is the wake-on component. The design of the wake-on component is achieved through some special design and modification in hardware. We will discuss this in detail in the section II-B. $TelosB$ [15] has been a popularly used general purpose mote for many sensor network deployments. Our $TelosW$ is upgraded from $TelosB$ by adding wake-on capability and energy meter. It also has the wake-on radio $CC1101$. We have kept the MSP430, as it has ultra low power consumption compared to other microcontrollers. MSP430 can also wake up fast from sleep mode (only about 6 μs). This microcontroller feature is beneficial for the wake-on operation. $TelosW$ also has an integrated design, combining programming, computation, communication and sensing onto one device. Users can program it through standard interface USB connector, and can also power it up through this.

$TelosW$ has a 20 pin IDC expansion header that includes general digital I/O, 0-5V,0-2.5V and 0-20mA analog input, UART, SPI interface,and I2C interface. The set of 16 pins of this header has the same function as $TelosB$. So all the sensor boards that are interfaced with $TelosB$, can also be used with $TelosW$. There are two kinds of analog voltage input on $TelosW$. One is the wake up analog input. Users can configure its wake-on threshold. If the input voltage is higher(or lower) than this threshold, it will trigger an interrupt, causing MCU to wake up. The other kind of analog voltage input is the normal input voltage without wake up functionality.

B. Wake-On Sensor Design

In this section we describe in detail how the wake-on capability is enabled in hardware. A typical sensor node collects ambience environmental data such as temperature, humidity, light intensity, acceleration, motion etc. On these nodes the microcontroller has to be ON all the time during sensing task. However, in different situations it is not necessary at all to sample all the sensors all the time. For example, when collecting infrasonic data for earthquake detection, most of time the signal level is not changing much. But what is actually needed is to detect some specific signal levels for specific events. The meaningful data for collection is the one which is different from normal level. If the infrasonic signal is converted into voltage signal, changes of infrasonic signal means changes in output voltage. Then this change will wake up the microcontroller. Realization of the need of this useful feature leads us to the wake-on hardware design.

The basic idea of wake-on ADC is as follows. Some components can detect the level of input signal, whether it is higher or lower enough to trigger an event (for waking up MCU). There are three key desired properties for such components. First, such component should work by itself without involvement of microcontroller. Second, the threshold level for triggering interruption should be configurable by the microcontroller. Third, the interruption should have two ways to be triggered: input higher than the high threshold, or input lower than the low threshold. Also the energy consumption of such components should be as low as possible.

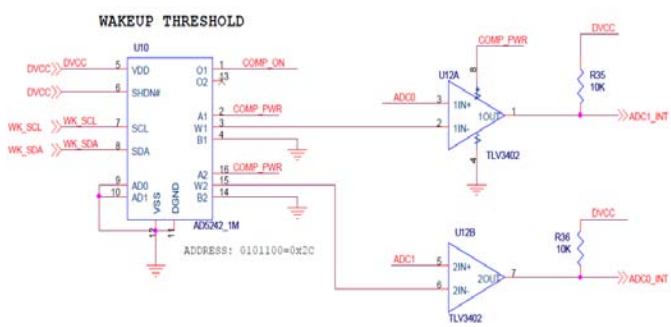

Fig. 4. Wake-On Design

Wake-On Circuit: The schematic of the wake-on design is shown in Figure 4. On $TelosW$ we have integrated the wake-on components with all the desired key properties. We have used two chips to build the wake-on component: Analog Devices AD5242 [1] and Texas Instruments TLV3402 [5].

The TLV3402 is a power comparator with only 470 nA supply current per channel. This makes it ideal for typical battery powered application. This device is used for comparing two voltages, say A and B. If input A is higher than input B, the output is high level voltage (about 3.3 V). Otherwise, the output is low level voltage (about 0 V). If the input A is fixed to a configured level, then the change in output can be used as a signal, denoting the change of input B with respect to A. This change in output can be used as an interruption to the microcontroller. In our design, one of the input of TLV3402 comparator is connected directly to the analog input on 20-pin expansion header. The other input is connected to one output of the AD5242.

The AD5242 is a dual-channel, 256-position, digitally controlled variable resistor device. It has 2-wire I2C interface, which makes it easy to communicate with the microcontroller. Using this device, we can get a fixed output voltage to TLV3402 as a comparison voltage. For every channel, there are three pins, A, W and B. A and B are two terminals of the resistor, W is the wiper terminal. The value of resistor from A to B is 1M Ohm. Commands can be sent from microcontroller in order to configure the resistor value between terminals W and B. In our design, we have connected B to ground, A to DVCC (the input of 3.3 V) and W works as voltage output. In this case, the value of output voltage from AD5242 can be configured by the microcontroller through I2C interface.

As shown in Figure 4, the left part of this schematic is AD5242, which controls the wake-on voltage level. The right part is TLV3402 which can generate interruption to the microcontroller. There is one observable feature that the power of TLV3402 does not come from the battery, but from an output pin of AD5242. This pin is an output pin which is available to drive digital loads, gates, LED drivers, analog switches, etc. This output can be controlled by I2C interface, which means if we don't want to use the wake-on feature on $TelosW$, the wake-on capability will be disabled by sending command from microcontroller. Then no more energy will be spent on this component.

The input to the wake-on ADC components are $ADC0$ and $ADC1$. $ADC0$ is connected with one of the 20-pin expansion header. So if this pin is connected with external sensors, it can detect the change in signal level from that sensor. For example, if it is connected to infrasonic sensor for detecting explosion, if explosion happens it will trigger interruption. $ADC1$ is connected to light sensor and also to one of the 20-pin expansion header. Since one ADC cannot intake two inputs, we add a 0 ohm resistor as a jumper connecting to light sensor. If the pin on the header is used, this resistor should be removed.

Wake-On by Vibration: Besides temperature sensor, humidity sensor and light sensor (as on $TelosB$), $TelosW$ has another on-board sensor, the accelerometer ADXL345. The ADXL345 [2] is a small, thin, low power, three-axis accelerometer with high resolution (13-bit) measurement of ±2g ±4g ±8g ±16g acceleration. The digital output data is formatted as 16-bit, and is accessible through either an SPI (3-wire or 4-wire) or I2C digital interface. In our design, we use I2C as the interface for communicating with the microcontroller. This accelerometer provides two interruptions with eight interruption sources (configured by the software). If the vibration is bigger than the preconfigured level, one of the interruptions will be trigged. Then if the vibration stops for a while, the other interruption will be sent to microcontroller. In this way, the accelerometer can tell the microcontroller when to wake up and when to go back to sleep again.

C. Energy Meter

One more important feature of $TelosW$ is the on-board energy meter.

Hardware: The on-board energy meter of $TelosW$ uses the idea and design of $iCount$ [7]. The work of $iCount$ proposes a hardware design that enables energy metering for free. This can be implemented on power limited sensor nodes to track

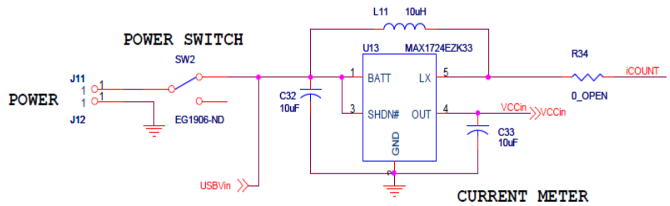

Fig. 5. Energy meter

their energy consumption from time to time. Basically iCount measures energy usage by counting the switching frequency of the regulator. The principle is that the relationship between the load current and the switching frequency is linear.

Figure 5 shows the on-board energy meter circuitry of $TelosW$. The energy meter circuit requires a voltage regulator. On $TelosW$ we add voltage regulator MAX1724 [4] which makes the power supply more stable. The MAX1724 is a compact, high-efficiency, step-up DC-DC converter. It features a low 1.5 μA quiescent supply current to ensure the highest possible light-load efficiency. The input power can come from two sources, the battery, or the USB. Using the regulator, the range of input voltage can be extended to 0.8 V to 5.5 V. As a comparison, $TelosB$ can only work when input voltage is higher than 1.8 V. This may potentially mean $TelosW$ can utilize more energy from the battery than $TelosB$ can.

As in Figure 5, J11 and J12 are battery connectors. Between them and the regulator, we use a switch that can turn off the power supply. In our design, the LX pin of MAX1724 is connected with port 2.7, the external clock for the Timer A subsystem of MSP430. Then the timer can count frequency of the switcher. The hardware counter works automatically and does not need the involvement of microcontroller. In order to use the energy meter correctly, the input voltage of the regulator is required. To achieve this, a wire is connected between BATT pin of the regulator and the 0-5V ADC input on 20-pin expansion header. Usually, the voltage range of the battery is 0-3V. So our 0-5V input range is enough for battery voltage testing.

Calibration: The relationship between the energy meter counter value (the regulator switching frequency count) and the load current is linear. This principle is the core of real time measurement of node energy consumption. But without proper calibration the energy measurement is not possible. The calibration parameters for the relation between counter value and load current varies with input voltage. For this purpose we have used ADC to detect the voltage input. However, the voltage of battery will not stay same all the time. Therefore depending on the present input voltage everytime we have to do the calibration. In our calibration, we use resolution of 0.1 V for input voltage, meaning that we calibrate the energy consumption for every 0.1 V variation in input voltage.

Through a large number of experiments with varying load and for varying input voltage, we have calibrated the raw energy meter reading to reflect the current consumption. We have fitted the raw energy meter reading and the actual current draw reading in the model of linear relationship (in log scale) between frequency count (by energy meter) and current. The set of calibration equations for varying input voltage are shown in Figure 6.

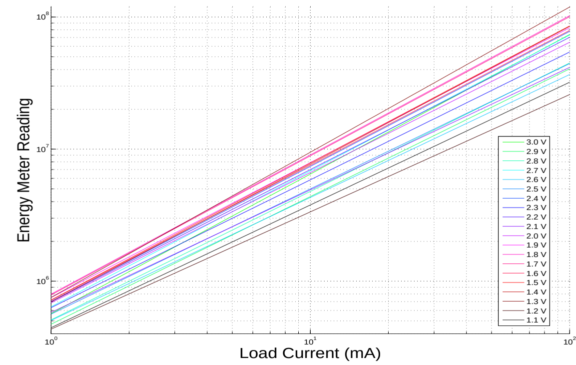

Fig. 6. Calibration model of energy meter for varying input voltage

Validation: The calibration parameters are obtained through a number of trials and experimentation. In this section we validate the correctness and robustness of our calibration for measurement of real time energy consumption.

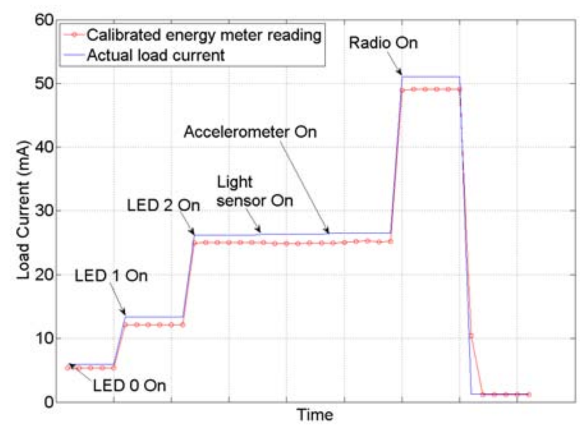

Fig. 7. Validation of on-board energy meter measurement

The validation experiment is set up as follows. The $TelosW$ node operates with varying load. Through time more components are activated, thus generating gradually increasing load. During the experiment we observe the actual current consumption by attaching a multimeter to the mote. The energy meter counter value is sent periodically through the serial forwarder to a PC. The current consumption measured by the on-board energy meter is visualized on a GUI and stored as raw data. Then we match the actual current value with energy meter current value, in order to validate the correctness of our calibration scheme. We show an example run of validation experiment in Figure 7. The application run by the node gradually increases the load. This is done by gradually turning on different components like LED0, LED1, LED2, light sensor, accelerometer, radio. This makes gradual increase in current draw. After experiment we calibrate the raw energy meter reading and compare it with corresponding multimeter reading. Through all the validation experiment run we observed that the measurement of energy meter is fairly accurate for all range of loads. This validates the correctness and robustness of $TelosW$'s on-board energy meter as an indicator of real time energy consumption.

Energy Consumption of Different Components: Since a real-time energy meter is integrated on the platform, it is possible to get energy consumption of each part on board in real-time. The test result is presented in Table I. The input voltage was fixed to 3.0 V.

Components	Energy Consumption
Red LED	4.72 mA
Blue LED	7.69 mA
Yellow LED	15.09 mA
Accelerometer	273 uA
Temperture Sensor	79 uA
Radio	24.4 mA
1ms Timer	1266 uA
10ms Timer	154 uA
100ms Timer	46 uA
1s Timer	34 uA
ADC Read Every 1 ms	3670 uA
ADC Read Every 10 ms	1610 uA
ADC Read Every 100 ms	395 uA
ADC Read Every 1 s	64 uA

TABLE I
ENERGY CONSUMPTION OF DIFFERENT COMPONENTS

III. WAKE-ON DESIGN ANALYSIS

The wake-on capability of $TelosW$ enables energy savings and release MCU from high load in two ways: wake-on sensor and wake-on radio. The wake-on sensor capability lets the MCU sleep, while the sensors (light, accelerometer, external sensor) trigger interrupt to wake MCU up. Therefore the MCU doesn't have to sample the ADC until interrupted by only meaningful events. The wake-on radio lets the MCU sleep while radio duty cycles on its own. In wake-on radio CC1101, the low power listening mode is hardware enabled. So the MCU can be waken up by radio only on receiving some radio message. This again adds to more energy savings, and let MCU focus on other tasks rather than duty-cycling radio. In this section we have done experiments in order to assess and validate the savings of energy due to wake-on sensor and wake-on radio. All the results support the capability of energy efficient wake-on property of $TelosW$.

A. Sensor Wake-On

In this section we analyze and validate the energy savings due to sensor wake-on. We have performed two experiments in order to validate the energy efficiency and detection accuracy due to sensor wake-on.

The first experiment is set up as follows. Two $TelosW$ nodes are programmed, one without sensor wake-on and one with sensor wake-on. Say node 1 is running without sensor wake-on mode (with continuous sensing) and node 2 is running with sensor wake-on mode. Both of them experience same events of change of light intensity. On detection of light intensity going below a threshold, both the nodes send notification message containing sampled data through serial forwarder. In addition to event message, both the nodes also send periodic message (every 30 seconds) with energy meter. The amount of energy consumption due to communication is made equal in both the nodes, ensuring fairness in comparison process.

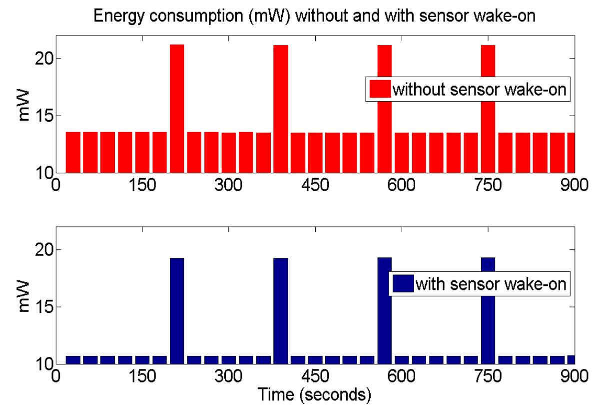

Fig. 8. Event detection and energy consumption without and with sensor wake-on

Figure 8 shows the average energy consumption in mW during each of the 30 seconds intervals. When no events occur the average energy consumption of node 1 (without sensor wake-on) is 13.5 mW, higher than that of node 2 (with sensor wake-on), which is 10.5 mW. During no event, the sensor wake-on capability saves energy from more idle MCU. Without sensor wake-on the MCU is always awake for sampling the light sensor. For sensor wake-on the light sensor is on, but MCU is not interrupted because of no event occurrence. The average power consumption goes high only during the interval where light events occur. It was also observed that both detect same set of events. These prove that sensor wake-on design detects all the events correctly and precisely, while saves more energy than without sensor wake-on. So sensor wake-on is advantageous in energy savings during the entire time when no events occur. But the MCU without sensor wake-on will keep sampling even if no events happen.

It is also worth noting that besides energy savings, the sensor wake-on has one more important advantage. Without wake-on, the event is detected only when the MCU does sampling. So the detection delay and even the success ratio of detection are dependent on the sampling rate. But with sensor wake-on the MCU is interrupted through hardware. So every meaningful event can be detected by MCU successfully and with lower detection delay. This is validated from the results from a second experiment as shown in Figure 9. In this experiment, there are two $TelosW$ nodes, say node 1 without

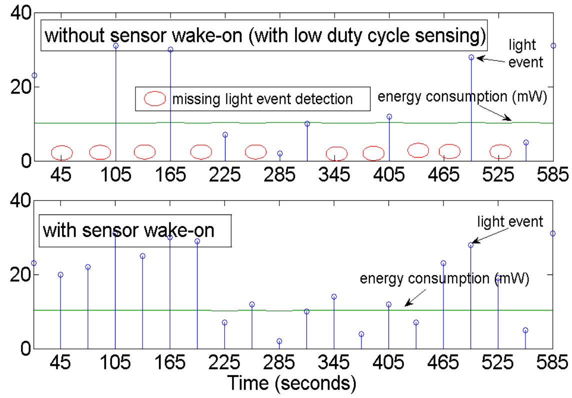

Fig. 9. Event detection accuracy without and with sensor wake-on

sensor wake-on and node 2 with sensor wake-on. Same as last experiment, they send periodic energy meter reading every 30 seconds through serial communication. Also they send one light reading message on each occurrence of light event, which happens every 30 seconds. Node 1 (without sensor wake-on) reduces its energy consumption by reducing the sampling rate to 0.5 Hz. From the results in Figure 9, node 1 is able to reduce energy consumption rate to same as that of node 2 (with sensor wake-on). But this comes in expense of a number of missed event detections, while node 2 doesn't miss any event. All the experimental results show that the sensor wake-on capability of $TelosW$ enables low energy consumption rate while providing reliable event detection.

B. Radio Wake-On

In this section we analyze and validate the energy savings due to Wake-On Radio (WOR). Wireless communication in sensor networks is highly energy consuming, making it one of the main source of energy drain. The CC1101 radio on $TelosW$ supports WOR mode, that can save significant amount of energy of communication task. WOR saves energy by using hardware based low power listening and more idle MCU. Besides sensor wake-on, the WOR operation of radio allows MCU to sleep more and wake up only on meaningful events or message reception. We have conducted experiments to show the energy efficiency of radio wake-on. In the experimental setting there are three pairs of sender-receiver. Each pair uses one of the following radio communication patterns: always on, duty cycled (using X-MAC [6]), and Wake-On Radio (WOR). In each pair, the sender periodically sends message to the receiver. In always on mode, the receiver radio is continuously listening for incoming packet. In duty cycled mode, MCU of the receiver duty cycles its radio. In the WOR mode, the radio duty cycles on its own. It wakes up the MCU if any message is received. The duty cycle is kept same for X-MAC and WOR for fair comparison.

Figure 10 shows the difference of energy consumption between X-MAC mode and WOR mode. The WOR mode is observed to be more energy efficient than X-MAC. In WOR

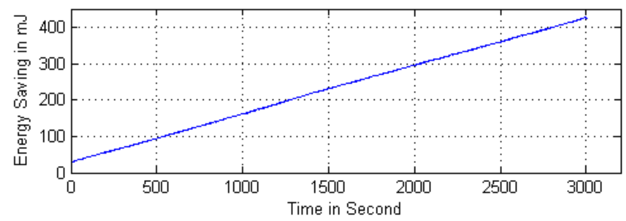

Fig. 10. Savings in energy consumption with WOR

the radio duty cycles on its own without intervention of MCU. This leads to estimated 12.7J energy saving per day due to our experiment, which is about four percent energy efficiency than X-MAC. The results prove energy efficiency of radio wake-on (WOR). Since $TelosW$ is equipped with both wake-on sensor and wake-on radio, it has the capability to reduce the energy consumption of sensor network applications by significant amount. Furthermore, WOR is not only designed for saving energy, but also relieves microcontroller from other tasks than computing. As an example scenario where it is not required to send data through radio very frequently, but have to get sensor reading with high sampling rate and to process these data (such as compression). In this case, if non WOR mode is in use, the microcontroller will have to duty-cycle radio and also sample sensor, which may create high load on it. However WOR mode will let the microcontroller only focus on sampling and processing sensor data, thus reducing load.

C. Software Architecture of Wake-on Design

In traditional design without wake-on feature, program is usually controlled by timers. Since $TelosW$ is a wake-on platform, program is controlled mostly by events. After the microcontroller is booted up, it configures every component, such as accelerometer, wake-on threshold and radio driver. Then it falls to sleep until external event triggers.For example, when environmental light intensity becomes low, it triggers an interruption to the microcontroller. After waking up from this interruption, the microcontroller starts a timer for light sensor sampling and sends data through radio. If the reading from light sensor exceeds a certain level (meaning light intensity becomes high again), light sampling timer stops. This changes the microcontroller status from active mode to sleep mode. Figure 11 shows the architecture of software with wake-on feature. Most time MCU is in sleep mode and wait for interruption from either sensors or radio. If these events happen MCU will sampling sensor with a suitable sampling rate or sending the data through radio or serial port.

IV. Distribution of Energy Consumption

The on-board energy meter hardware and the relevant driver (with proper calibration) of $TelosW$ provide the facility of tracking real-time energy consumption of each node. This can be utilized through network wide energy data collection. This has motivated us to track the distribution of energy consumption across the whole network from time to time. It

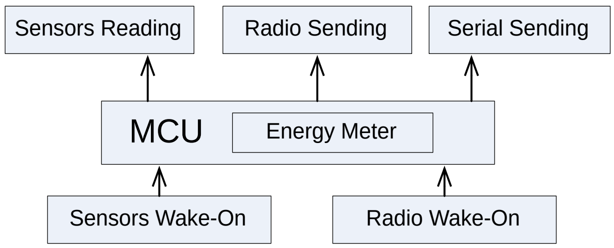

Fig. 11. Software Architecture

gives precise real-time view of energy state of the nodes across the network of any size.

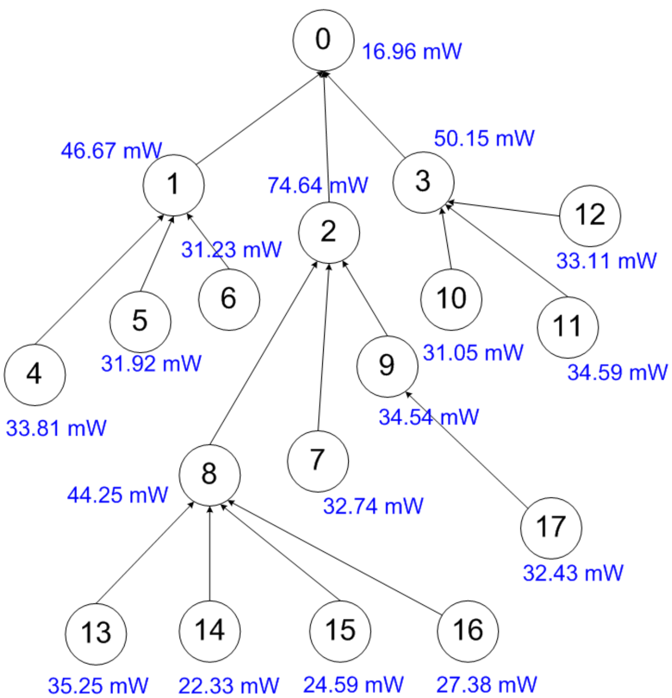

Fig. 12. Network topology and distribution of energy consumption

We have used the network topology as shown in Figure 12. In the experiment each node sends its energy meter reading every 2 seconds. The radio on transceiver operates on low power listening mode. This significantly reduces the energy consumption on the transceivers. The energy consumption rate (in mW) of the nodes in the routing tree supports the relation between node energy consumption and its position in network. The results prove the relevance and correctness of the energy meter.

The pattern of energy consumption rate of nodes in the network is presented in Figure 12. It is verified that the nodes closer to the sink have more energy consumption rate. Among the 1-hop nodes from the base station, node 2 has the highest energy consumption rate. This is because it forwards traffic from more number of nodes (8 nodes) than node 1 or 3 (each of which forwards traffic from 3 nodes). Also among the 2-hop nodes, node 8 has most energy consumption rate, because it has most number of child nodes (4 nodes).

The sink node in Figure 12 has the lowest energy consumption. That is because the sink node only receives packets, and does not transmit packets. According to CC1101's datasheet, it will consume 15.6 mA when in receive mode while 32.3 mA when in transmit mode (at 250kBaud data rate and 10dBm output power). The energy consumption of serial communication is not included, since the components of serial communication on $TelosW$ are powered up by USB port.

V. Multiple Data Rate

The CC1101 radio also supports multiple data rate, besides multiple frequencies and multiple power levels. $TelosW$ has following programmable data rates available: 1.2 KBaud, 2.4 KBaud, 10 KBaud, 38.4 KBaud, 76.8 KBaud, 100 KBaud, 150 KBaud, 250 KBaud and 500 KBaud. This feature can be used for adapting the transmission range in a network. We have done testing on the range in an outdoor campus environment. The sender $TelosW$ mote, placed on top of a 2.5 ft high pole from the ground, periodically sends beacons. The LED of receiver $TelosW$ mote indicates any reception of packet. The receiver is moved in the test in order to detect the reliable transmission range. Data rates used are: 1.2 KBaud, 10 KBaud, 38.4 KBaud, 100 KBaud, 150 KBaud and 250 KBaud, all with Manchester coding. Figure 13 shows the observation from the experiment. Although the range observed is not the maximum that can be achieved with CC1101 ideally. This was because of the factors of test environment, like interference, reflection etc. But the test results are fair, and also support the trend of increase in transmission range with decrease in data rate.

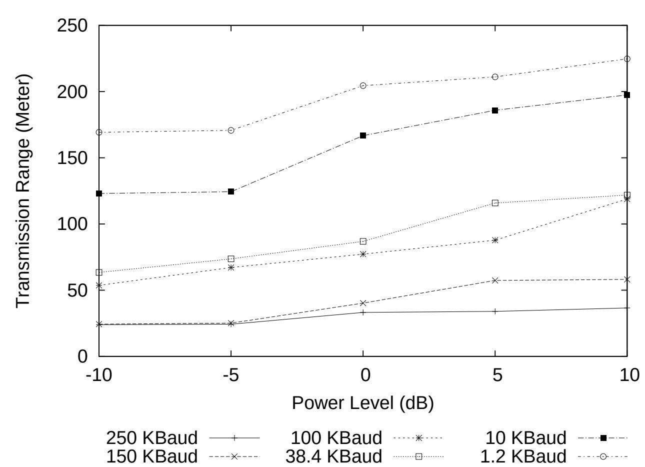

Fig. 13. Transmission Range with Different Data Rate

From this figure, we can get the conclusion that lower the data rate, higher is the transmission range, and vice versa. Lower data rate increases the transmission time but also provides higher link reliability at same distance. Hence there is a trade off between data rate and energy consumption.

VI. Related Work

In recent years there have been significant developments both in sensor hardware platform and in distributed sensor network algorithms. Recent time has witnessed different kinds of sensor platforms. Some platforms are application specific with special commodity hardware. There are also some widely applicable sensor platforms. Some of the commonly used sensor platforms are TelosB [15], MicaZ, iMote2. There are also available sensor motes with energy harvesting capability such as Trio [8], Heliomote [12], Twin-Star [19]. In the existing sensor platforms the MCU has to wake up to sample the sensor data and then detect event, if any. But in our developed platform $TelosW$, instead of MCU reading sensors to detect events, sensors will wake up MCU upon event occurrences. The capability of the events to wake up the MCU can be preconfigured as desired, thus adding flexibility. This novel wake-on design changes the way that event detection is handled on sensor platform. The work in [13] proposes an active RFID design which supports asynchronous monitoring of exceptional events and is similar as our sensor wake-on. The work in [9] proposes a simulation design of radio triggered hardware component for power management. It is not a actual hardware design.

Real time energy metering of nodes in a sensor network has been an important research issue. Most of the previous works use software techniques for predicting the real time energy consumption based on certain model [14]. But software prediction of energy consumption doesn't guarantee to give accurate result due to various issues, like correctness of model, hardware variations, energy leakage etc. Rather a hardware component is more suitable that can reliably measure the real time energy consumption. [11] proposes a dedicated energy monitoring and management hardware. The work by Dutta et al. [7] introduces iCount, an energy meter hardware that can measure real time current consumption for free (at almost no cost of extra energy). The principle is based on linear relationship between load current and switching frequency in a switching regulator. This energy meter is very much suitable for embedding on sensor node because of its reliability and negligible power consumption. Our developed $TelosW$ platform is equipped with an on-board energy meter that can reliably measure the real time energy consumption. Detailed calibration of raw energy meter reading is performed. The energy metering on $TelosW$ is robust to varying loads and varying input voltage.

VII. Conclusion

In effort to increase the energy efficiency of wireless sensor networks we have proposed an ultra-low power wake-on sensor network. The wake-on sensor network is enabled by a new kind of sensor platform, named $TelosW$. It exploits the event based nature of sensor networks. The design of $TelosW$ consists of wake-on hardware that enables the MCU to wake up from sleep state, only to configurable events. $TelosW$ also includes the Wake-On Radio (WOR) hardware. These all enable event triggered sensing and communication, leading to true wake-on operation. $TelosW$ also has a precision energy meter. Using this energy meter it is possible to get in-situ insight of the energy states of the nodes in a sensor network. In this paper we have presented the hardware design of $TelosW$ in detail. Using the energy meter we have obtained the insight to distribution of energy consumption across network. The energy meter is also used to validate the energy efficiency of wake-on design. We have also evaluated the performance of the radio on $TelosW$. All the results support the correctness, robustness, stability and other properties of $TelosW$ for its wide range of applicability.

References

[1] *AD5242 Datasheet,Analog Device*. http://www.analog.com/en/digital-to-analog-converters/digital-potentiometers/ad5242/products/product.html.

[2] *ADXL345 Datasheet, Analog Device*. http://www.analog.com/en/sensors/inertial-sensors/adxl345/products/product.html.

[3] *CC1101 Datasheet, Texas Instruments, Available at*. http://focus.ti.com/docs/prod/folders/print/cc1101.html.

[4] *MAX1724 Datasheet, MAXIM*. http://datasheets.maxim-ic.com/en/ds/MAX1722-MAX1724.pdf.

[5] *TLV3402 Datasheet, Texas Instruments*. http://focus.ti.com/docs/prod/folders/print/tlv3402.html.

[6] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In *SenSys'06*, 2006.

[7] P. Dutta, M. Feldmeier, J. Paradiso, and D. Culler. Energy metering for free: Augmenting switching regulators for real-time monitoring. In *SPOTS'08 IPSN Conference Proceedings*, April 2008.

[8] P. Dutta, J. Hui, J. Jeong, S. Kim, C. Sharp, J. Taneja, G. Tolle, K. Whitehouse, and D. Culler. Trio: enabling sustainable and scalable outdoor wireless sensor network deployments. In *IPSN'06*, 2006.

[9] L. Gu and J. Stankovic. Radio-triggered wake-up capability for sensor networks. In *RTAS'04*, 2004.

[10] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. A. Stankovic, T. F. Abdelzaher, J. Hui, and B. Krogh. Vigilnet: An integrated sensor network system for energy-efficient surveillance. *ACM Transactions on Sensor Networks*, 2006.

[11] A. Hergenroder, J. Horneber, D. Meier, P. Armbruster, and M. Zitterbart. Distributed energy measurements in wireless sensor networks. In *SenSys'09*, 2009.

[12] K. Lin, J. Yu, J. Hsu, S. Zahedi, D. Lee, J. Friedman, A. Kansal, V. Raghunathan, and M. Srivastava. Heliomote: enabling long-lived sensor networks through solar energy harvesting. In *SenSys'05*, 2005.

[13] M. Malinowski, M. Moskwa, M. Feldmeier, M. Laibowitz, and J. A. Paradiso. Cargonet: A low-cost micropower sensor node exploiting quasi-passive wakeup for adaptive asynchronous monitoring of exceptional events. In *Sensys'07*, 2007.

[14] E. Perla, A. O. Cathain, and R. S. Carbajo. Powertossim z: Realistic energy modelling for wireless sensor network environments. In *Proceedings of the 3nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, October 2008.

[15] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling ultra-low power wireless research. In *IPSN'05 Conference Proceedings*, pages 364–369, April 2005.

[16] W. Song, R. Huang, M. Xu, A. Ma, B. Shirazi, and R. Lahusen. Air-dropped sensor network for real-time high-fidelity volcano monitoring. In *Mobisys'09*, June 2009.

[17] A. Wood, J. Stankovic, G. Virone, L. Selavo, Z. He, Q. Cao, T. Doan, Y. Wu, L. Fang, and R. Stoleru. Context-aware wireless sensor networks for assisted-living and residential monitoring. *IEEE Network*, 2008.

[18] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, , and D. Estrin. A wireless sensor network for structural monitoring. In *SenSys'04*, 2004.

[19] Z. Zhong, T. Zhu, T. He, and Z. Zhang. Leakage-aware energy synchronization on twin-star nodes. In *SenSys'08*, 2008.