doi:10.4236/wsn.2010.29084 Published Online September 2010 (http://www.SciRP.org/journal/wsn)

Sensors Dynamic Energy Management in WSN

Xianghui Fan, Shining Li, Zhigang Li, Jingyuan Li
College of Computer Science, Northwestern Polytechnical University, Xi'an, Shaanxi, China
E-mail: *fxianghui@vip.qq.com*

Received July 2, 2010; *revised August* 16, 2010; *accepted September* 12, 2010

Abstract

A wireless sensor node is typically battery operated and energy constrained. Therefore, it is apparent that optimal energy management is one of the most important challenges in WSN development. However, energy management requires in-depth knowledge and detailed insight concerning specific scenarios. After Carrying out a large number of experiments in precision agriculture, we find that it is the sensors that have never been concerned consuming the most energy of the node. In order to conserve energy and prolong the lifetime of WSN, we design and carry out a dynamic energy management strategy for sensors. The basic idea is to shut down all sensors' power when not needed and wake them up when necessary. Valuable conclusions are extracted and analyzed.

Keywords: WSN (Wireless Sensor Network), Precision Agriculture, Dynamic Energy Management, TinyOS

1. Introduction

A large number of intelligent micro-sensor nodes with sensing, processing and wireless communicating capabilities form wireless sensor network (WSN), which completes complicated tasks in some specific field, such as precision agriculture. Compared with old methods, WSN has significantly drawn extensive attention. It does not rely on fixed infrastructure and has many characteristics such as fast setup, strong survivability and so on [1]. It has been considered as a good scheme to conduct precision agriculture data collection and processing. In 2002, Intel has a project looking at how WSN can be used to improve grape production. They worked with agricultural scientists on a long-term deployment of WSN in a wine grape vineyard. By densely monitoring and analyzing they found the relationship between grape quality and climatic conditions. It has been proved that WSN could play a role in precision agriculture.

Just the same as other applications, energy constraint of sensor nodes is the major problem for precision agriculture. Data aggregation [2] and low power listening [3] algorithms are effective method to reduce energy consumption in normal wireless sensor networks. However, after a sufficient number of experiments we found that energy consumption in precision agriculture has some special issues. Generally speaking, in order to monitor the growth conditions of crops, one node has to connect with many sensors, such as Co2 sensor, air temperature sensor, air humidity sensor, light sensor, soil temperature/moisture sensor and so on. Although the sensors consume a large portion of the energy, we never pay any attention to this issue in our previous research. It is necessary to reduce the energy consumption of sensors. We design and carry out a sensor dynamic energy management (SDEM) to reduce energy consumption of sensors and extend network lifetime. The basic idea is to shut down sensors when not needed and wake them up when necessary [4,5]. The experimental results indicate that SDEM is an effective technique in reducing node energy consumption without significantly degrading performance.

The remainder of this paper is organized as follows. Section 2 gives the energy consumption of all parts of the sensors in precision agriculture. And we get a conclusion that the sensors consume most of the energy. The architecture of the SDEM is described in Section 3, and Section 4 reports the hardware and software design and some considerations about implementation. In Section 5, the actual deployment is described in detail, and finally Section 6 reports our conclusion and gives some directions on the ongoing work.

2. Sensors Energy Consumption Analysis

A sensor node has several major components: processor, memory, A/D converter, sensing unit and radio. Each node sleep mode corresponds to a particular combination

Copyright © 2010 SciRes.

of component power modes. In general, if there are N components labeled (1, 2, … , N) each with K_i sleep modes, the total number of node sleep modes is $\sum_{i=1}^{n} k_i$ [6]. However, from a practical point of view not all sleep modes are useful. Optimizing the key issue could achieve a noticed effect. In our case, the NPUnode used in precision agriculture consists of a processor Atmel2561, which has rich resource such as 256K bytes in-system programmable flash, 4K bytes EEPROM, 8K bytes SRAM and so on, a RF230 chip acts as the radio unit, an AT45DB041B chip acts as an extern memory. And the NPUnode has been equipped with six sensors, just as showing in **Figure 1**. Each component is controlled by the micro-operating system (TinyOS). **Table 1** enumerates the characteristics of all devices of the NPUnode.

Sensor transducers translate physical phenomena to electrical signals, and are classified as either analog or digital devices depending on the type of their output. There are several sources of power consumption in a sensor, including 1) signal sampling and conversion of physical signals to electrical ones, 2) signal conditioning, and 3) analog to digital conversion. Given the diversity of sensors there is no typical power consumption number. In general, however, passive sensors such as temperature sensors etc., consume negligible power relative to other components of sensors, array sensors such as Co2 sensors can be large consumers of power. As shown in **Table 1**, the current value of Co2 sensor is 30mA, sometimes even reaches the max value 150mA. And other sensors also have power dissipation.

In accordance with **Table 1**, we draw a figure of energy consumption of major components. Just as shown in **Figure 2**, the sensors energy consumption is far more than other components. According to electric power formula $W = Pt = UIt$, we know that reducing the running time is the only way to decrease energy consumption while not change the sensors characters. T is the data collection period, while T_{woking} is the working time of sensors and T_{idle} means the period that sensors is in the idle state. And the relationship between them is $T = T_{working} + T_{idle}$ and $T_{idle} >> T_{working}$. Data collection period is a long time in agriculture application. For example, the nodes we have deployed in greenhouse in YangLing only need to report the sensor data once an hour. This means that the sensors only need to work for several minutes per hour, so they should be turned off when they are idle to save energy.

3. Sensors Dynamic Energy Management

Nodes energy consumption is application specific. In precision agriculture the sensors consume the greatest portion of the node energy. The reason is that one node is always connected with several sensors to monitor plant growth environment. However, the most time of sensors is in idle state and waste a lot of energy. For the purpose of saving energy of nodes and extending the life of WSN, we design the Sensors Dynamic Energy Management (SDEM) strategy which turns on the sensors when the node receives an acquisition command from root and turns it off when the sensors are in the idle state. The principle of SDEM is showed as **Figure 3**.

3.1. Hardware Design

SDEM needs independent hardware design supporting. The NPUnode supports power with DC 5V or 3.6V Nickel-Hydrogen Battery. And the sensors used to monitor the plant growth environment need different voltage. A power supply of DC3.0V is designed for light sensor, soil temperature sensor, temperature and humidity sensor. Co2 sensor and soil Moisture sensor power are supplied with DC5.0V. The hardware design is described as in **Figure 4.** XC6221B and TPS61202 chips which are connected with ATmega2561 I/O pins translate battery power to DC3.0V and DC5.0V to meet above

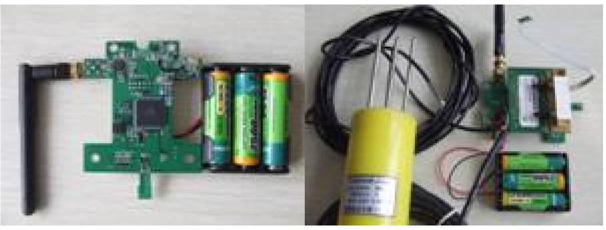


Figure 1. The NPUnode and equipped sensors.

Table 1.The devices of the node and their characteristics.

Device	Type	Characteristics
Processor	ATmega 2561V	256K Bytes Flash, 4K Bytes EEPROM, 8K Bytes Internal SRAM, Power :2.7~5.5V,10mA
Radio	RF230	Power: Voltage: 1.8V~3.6V, SLEEP: 0.1 µA, RX: 16 mA, TX: 17 mA; Fast Power-Up Time < 1 ms
Flash	AT45DB041	Power Supply: 2.7V~3.6V, 4 mA Active Read
Temp and Humi Sensor	SHT11	Temperature range:-40℃ to +123.8 ℃;Humidity range: 0 to 100% RH; Power : 3V, 0.5mA
Light Sensor	ISL29002	Range: 10,000lux~100,000lux;Power : 2.5V to 3.3V, <10mA
Soil Temp Sensor	PT1000	Range: -70°C to + 500°C;Power:3VDC, 3mA;Response time < 10s
Soil Mois Sensor	TDR-3	Range: 0～100%（m3/m3）； Power Supply: 4.5~5.5V DC, 50～70mA； Response time: < 1s
Co2 Sensor	GE/Telaire 6004	Range:0-2000ppm;Power: 4.3 VDC ~7.0 VDC, 30~150mA;Response time: < 2 min

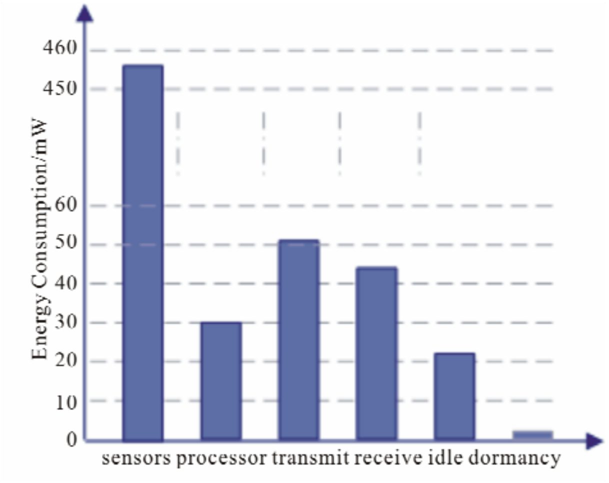


Figure 2. Energy consumption of node in precision agriculture.

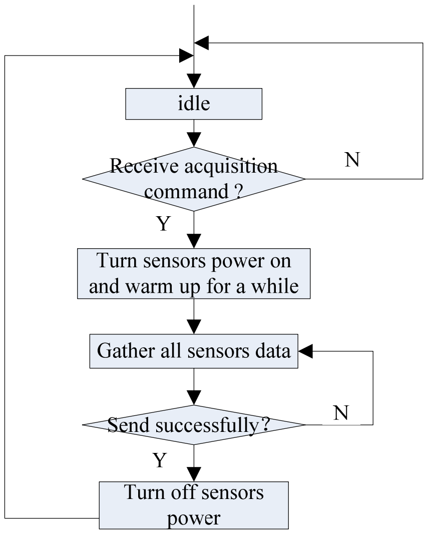

idle
Receive acquisition command ?
N
Y
Turn sensors power on and warm up for a while
Gather all sensors data
Send successfully?
N
Y
Turn off sensors power

Figure 3. Principle of sensor dynamic energy management.

demand. Inputting L level signal for I/O pins we can provide power for sensors. On the other hand, we can turn off the sensors power supply by inputting H level signal for I/O pins. It is easy to implement in TinyOS.

3.2. Software Design

The software system considered here is based on TinyOS 2.1 [7] which is an embedded operating system especially for WSN applications. The extreme power limitation of nodes forces the operating system to take very different approaches than traditional computing classes. TinyOS is implemented using NesC programming language—a dialect of C programming language. It integrates with model of components/module and events driven model. Over the past few years, TinyOS has grown from a small research project to dominant operating system for low power wireless sensor networks.

At a high level, TinyOS provides three things to make writing systems and applications easier [8]. 1) a component model, which defines how you write small, reusable pieces of code and compose them into larger abstractions; 2) a concurrent execution model, which defines how components interleave their computations as well as how interrupt and non-interrupt code interact; 3) application programming interfaces (APIs), services, components libraries and an overall component structure that simplify writing new applications and services. The HplAtm256-GenerallIOC is the most important components to design the SDEM. It exposes the ATmega256's 53 digital I/O pins as 53 GeneralIO interfaces, hiding the slightly different instruction sequences needed to perform some operations on some I/O pins. Some I/O pins can be set atomically in a single assembly instruction. On the purpose of turn on/off the sensors power, we need to use the interface GeneralIO to enable/disenable the pins of Atmel2561 which connect with XC6221B and TPS61202 by inputting Low/High level signal. The GeneralIO interface offers commands to configure, read and write a typical microcontroller digital I/O pin. Firstly, after the OS initializes all needed components and booted successfully, we call GeneralIO.makeoutput() to make the pins as a controller. Then the NPUnode enters a

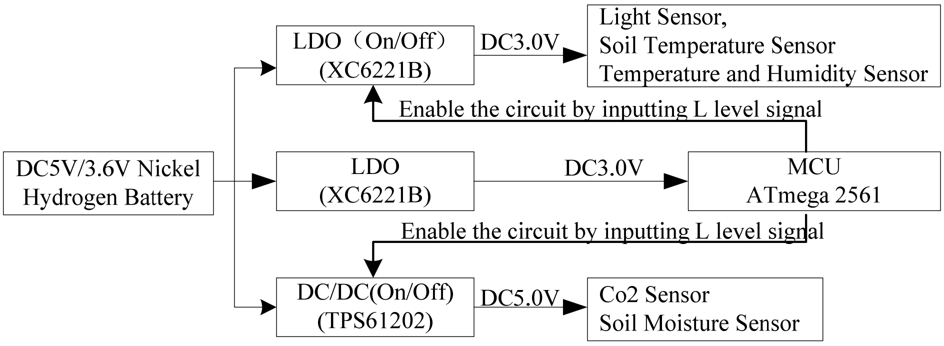
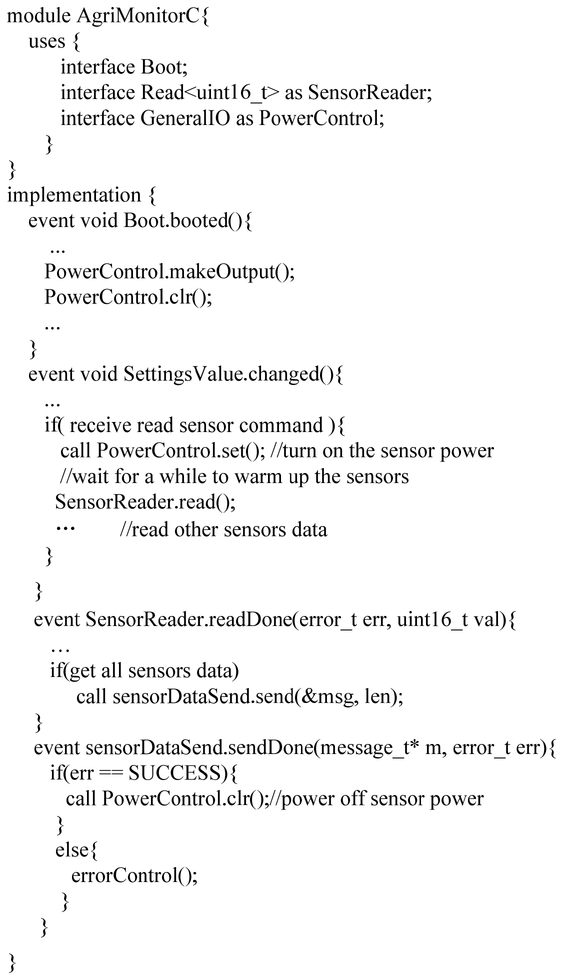


Figure 4. The hardware design of SDEM.

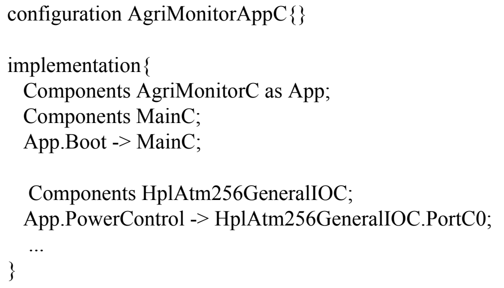
Copyright © 2010 SciRes.

ready state waiting for the command from root. Once the node receives the sampling command, it then calls the command GeneralIO.set() to turn on the sensors power and waits for 2 minutes for warming up the sensors. After reading all sensors data and sending it back successfully, nodes calls GeneralIO.clr() to turn off the sensors power. The **Figure 5** and **Figure 6** show the implementation of SDEM.



```
module AgriMonitorC{
   uses {
        interface Boot;
        interface Read<uint16_t> as SensorReader;
        interface GeneralIO as PowerControl;
      }
}
implementation {
   event void Boot.booted(){
      ...
      PowerControl.makeOutput();
      PowerControl.clr();
      ...
   }
   event void SettingsValue.changed(){
      ...
      if( receive read sensor command ){
         call PowerControl.set(); //turn on the sensor power
         //wait for a while to warm up the sensors
        SensorReader.read();
        ···      //read other sensors data
      }
    }
    event SensorReader.readDone(error_t err, uint16_t val){
      …
      if(get all sensors data)
          call sensorDataSend.send(&msg, len);
    }
    event sensorDataSend.sendDone(message_t* m, error_t err){
      if(err == SUCCESS){
         call PowerControl.clr();//power off sensor power
       }
       else{
          errorControl();
        }
     }
}
```

Figure 5.The module implementation of SDEM.



```
configuration AgriMonitorAppC{}

implementation{
   Components AgriMonitorC as App;
   Components MainC;
   App.Boot -> MainC;

    Components HplAtm256GeneralIOC;
   App.PowerControl -> HplAtm256GeneralIOC.PortC0;
    ...
}
```

Figure 6.The configuration wired the needed components.

4. Experiment Results

We deployed NPUnodes in field crop production in YangLing, LuoChuan, AnSai in ShaanXi province, just as showing in **Figure 7**. The NPUnodes are equipped with Co2 sensors, temperature and humidity sensors, light sensors, soil temperature sensors and soil moisture sensors to collect important data for plant growth environment. The collected data is gathered at the edge of the field by a field gateway and further transferred via GPRS to a PC server for data analysis. Once something odd happened the server will send message via MMS to farmers. In addition to the agronomic experiment, we expect to gather data and statistics on the behavior of NPUnodes in real-world experiment. For energy-efficiency considerations, we use SDEM strategy in NPUnodes and they reported data only once per hour. To get the exact value of sensors, we need to wait sensors warm-up for two minutes. The rest of the time we turn off the sensor power to save energy.

From collected voltage data of NPUnodes, we make a comparison between the SDEM strategy and before methods. Just showing in **Figure 8**, we can give the conclusion that the SDEM strategy has significantly saved energy. After making in-depth research we find that the

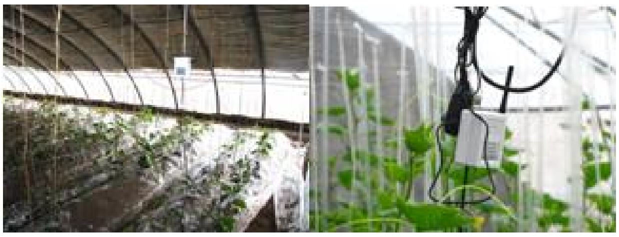


Figure 7. NPUnode deployed in greenhouses with six sensors.

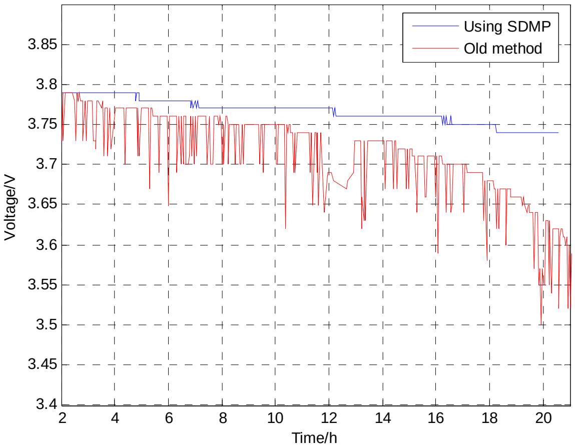


Figure 8. Lift time of sensors between SDEM and before method.

SDEM strategy almost has extended the NPUnodes life time from 1 month to 3 months. However, SDEM should save sensors energy 30 times than not using SDEM nodes, theoretically on the condition that NPUnode reports sensors data once per hour (all sensors only turn on for 2 minutes). The reason is that not only the sensors consume the energy, other components such as radio, processor etc. also consume the energy.

5. Conclusions and On-Going Work

In precision agriculture, nodes are always equipped with several sensors which consume a large portion of nodes energy, especially active sensors. In order to save energy and extend the lifetime of nodes, we design and implement the SDEM strategy based on our NPUnodes. The basic idea is to shut down devices when not needed and wake them up when necessary. And then we deployed our nodes in greenhouse. From the voltage data collected we give the conclusion that the SDEM strategy can significantly save energy of nodes, and extend the life time of nodes from one month to three month.

The nodes need continuous work for one year or more in precision agriculture. So the nodes need an unfailing supply of energy. Next time, we will make a serial research and supply solar power for NPUnodes. Maybe it is a way to solve the energy problem finally.

6. References

[1] S. M. Xiong, L. M. Wang, X. Q. Qu and Y. Z. Zhan, "Application Research of WSN in Precise Agriculture Irrigation," *International Conference on Environmental Science and Information Application Technology*, Wuhan, July 2009, pp.297-330.

[2] B. Krishnamachari, D. Estrin and S. Wicker, "Impact of Data Aggregation in Wireless Sensor Networks," DEBS'02.

[3] J. Polastre, J. Hill and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, Maryland, November 2004.

[4] A. C. Sinha, "Dynamic Power Management in Wireless Sensor Networks, Design & Test of Computers," *Proceedings of IEEE*, Vol. 18, No. 2, 2001, pp. 62-74.

[5] R. C. Luo, L. C. Tu and O. Chen, "An Efficient Dynamic Power Management Policy on Sensor Network," *Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, 2005, pp. 341-344.

[6] C. Lin, N. Xiong, J. H. Park and T.-H. Kim, "Dynamic Power Management in New Architecture of Wireless Sensor Networks," *International Journal of Communication Systems*, Vol. 22, No. 6, 2008, pp. 671-693.

[7] TinyOS Tutorials, Internet Available:
http://docs.tinyos.net/index.php/TinyOS_Tutorials

[8] P. Levis and D. Gay, "TinyOS Programming," Cambridge University Press, England, 2009, pp. 6-7.

Copyright © 2010 SciRes.