Analysis of power consumption for different sensor access modes

This item was submitted to Loughborough University’s Institutional Repository by the/an author.


Additional Information:

- This is a conference paper [© IEEE]. It is also available from: http://ieeexplore.ieee.org. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Metadata Record: https://dspace.lboro.ac.uk/2134/4142

Version: Published

Publisher: © IEEE

Please cite the published version.
This item was submitted to Loughborough’s Institutional Repository (https://dspace.lboro.ac.uk/) by the author and is made available under the following Creative Commons Licence conditions.

For the full text of this licence, please go to: http://creativecommons.org/licenses/by-nc-nd/2.5/
Analysis of power consumption for different sensor access modes

Zhenhuan Zhu and Shuang-Hua Yang

Abstract—This paper focuses on analysis of the power consumption of wireless sensor nodes with heterogeneous multi-sensors. The power consumption is investigated in three sensor access modes: serial access mode, parallel convert mode, and event driven mode. The mathematical models of the power consumption in these sensor access modes are established and evaluated in the paper. In terms of the solutions of wireless sensor nodes provided by Crossbow and Jennic, etc., a general framework of a wireless sensor node is proposed as well. The work can be used to guide the design of control system of a wireless sensor node.

KEY WORDS: wireless sensor nodes, sensor access modes, power consumption, energy awareness.

I. INTRODUCTION

Reducing power consumption is very important in the design of a wireless sensor system. It is because sensor nodes are battery driven and hence operate on an extremely frugal energy budget. Further, they must have a lifespan on the order of months to years, since battery replacement is not an option for networks with thousands of physically embedded nodes. The network lifespan can be maximized only by incorporating energy-awareness into every stage of wireless sensor network design and operation, thus empowering the system with the ability to make dynamic tradeoffs between energy consumption, system performance, and operational fidelity [9]. In general, the research to reduce power consumption is categorized into two. One is to look for an optimal solution based on wireless sensor networks (WSNs). Many routing, power management, and data dissemination protocols have been specifically designed for WSNs where energy awareness is an essential design issue [1]. The other is to design a wireless sensor node with its power consumption as low as possible.

Methods, which can significantly reduce power consumption in the design of a wireless sensor node, are concerned with selecting the electronic components with the lowest supply voltage and current, and using an optimal control scheme to manage work modes among all components inside the sensor node. The functions of an optimal scheme includes making microprocessor enter a sleep mode while the node is in an idle state, adjusting the transmitting power of the RF module according to the requirement, adopting optimal work modes with energy-awareness, and so on. The paper aims to investigate the relationships between power consumption and the work modes of driving heterogeneous sensors. Section 2 introduces the framework of a wireless sensor node. The general architecture of a wireless sensor node is described in Section 3, and the functions of all components inside the architecture are depicted separately. Section 4 discusses the relationships between power consumption and sensor access modes, and establishes corresponding mathematical models. Section 5 provides the evaluation to three sensor access modes. The conclusion is given in Section 6.

II. FRAMEWORK OF A WIRELESS SENSOR NODE

In the viewpoint of software system development, the framework of a wireless sensor node consists of such subsystems as communication, sensing, and calculation. Communication subsystem includes a RF module (RFM) for receiving and transmitting packets and implementation of communication protocols. For example, RFM in a MICAz node produced by Crossbow, shown in Fig. 1, is CC2420, a chip provided by Chipcon [8], which implements the physical layer of IEEE 802.15.4 protocol. The software part of the communication subsystem implements the remains of ZigBee and IEEE 802.15.4 protocols. Sensing subsystem consists of peripheral sensors, corresponding control circuits, and interface programs for driving sensors. For different applications, a lot of heterogeneous sensors are applied to detect different environment information, such as temperature, pressure, gas, color, and so on. Control circuits are used to implement interfaces with peripheral sensors. The interfaces often follow standard bus specifications such as I²C, UARTs, etc (see Fig. 1). Calculation subsystem usually consists of a microprocessor and a micro operation system that is suitable for the embedded systems, such as BOS [6], TinyOS, Bertha, SensorWare and Eyes [2]. In general, user’s application programme is loaded as a part of calculation subsystem.

According to the framework of Jennic node shown in Fig. 2, it is easy for application developer to understand the principle of a wireless sensor node. Communication function, sensor access, system control and management are converted into the operations of a serial of application programming interfaces (API). In other words, user’s application consists of a serial of calls based on IEEE 802.15.4 stack API. Accessing sensors can be implemented by calling the integrated peripherals API. In order to get a request from other nodes and to exchange data between tasks, event mechanisms are provided. All tasks mentioned above are managed by the BOS (basic operation system). The BOS is a simple task

Mr. Z. Zhu is a PhD student in the Department of Computer Science, Loughborough University, LE11 3TU, UK. E-mail: Z.Zhu@lboro.ac.uk.

Professor S. H. Yang is with the Department of Computer Science, Loughborough University, LE11 3TU, UK. Email: S.H.Yang@lboro.ac.uk.
scheduler and controls the execution of the ZigBee stack and user tasks. The ZigBee stack is implemented as three tasks - the Network layer, the APS (Application Sub-layer) and the ZDO (ZigBee Device Objects). The MAC sub-layer that sits below the ZigBee stack runs in interrupt context, as do events generated by the on-chip peripherals [6].

![Diagram](image1)

**FIG. 1 Framework of a MICAZ node [3]**

Therefore, the framework of a wireless sensor node, based on either SoC architecture or components structure, includes a microprocessor, a RF module, and one or more homogeneous or heterogeneous sensors generally. Communication protocol is implemented by the co-design both software and hardware. A micro operation system is used to monitor a whole node and to run the user’s application.

![Diagram](image2)

**FIG. 2 Framework of a Jennic node[5]**

### III. GENERAL STRUCTURE OF A WIRELESS SENSOR NODE

A general architecture of a wireless sensor node is abstracted and shown in Fig.3, which is derived from the solutions provided by such chip manufacturers as Chipcon, Jennic, Crossbow, Microchip, and so on.

It might be more reasonable to divide RF module as two components Rx and Tx because single microprocessor does not work for two components at the same time. In order to make the structure of the model more concise, the functions of calculation, control, and management are integrated in the process module. In this model, sensors are either homogeneous or heterogeneous. Links between two modules are described by two kinds of lines. The solid line represents data transmission and the broken line indicates the command from the main module (the process module) and the state information from the slave module i.e. sensors or RFM. The process module works in two states. One is an active state when the microprocessor is working. The other is a sleep state when the microprocessor stops work and power is reduced to a threshold value only to keep the information of RAM and registers inside the node. Power consumption of the process module is largely different in the active or sleep states, and the difference of power consumption also exists in the heterogeneous sensors. The differences of converting time among heterogeneous sensors are very large. For instance, an electrochemical sensor, the Compact-S provided by company Monox, spends less than 40s from starting to finishing detection. The transmitting time between heterogeneous sensors and the process module is different because different transmitting methods are used such as A/D conversion, serial and parallel data transmission.

![Diagram](image3)

**FIG. 3 General framework of a wireless sensor node**

### IV. RELATIONSHIPS BETWEEN POWER CONSUMPTION AND SENSOR ACCESS MODES

Modeling a control system with communication and computation modules can be applied to either an industrial communication network [7] or a wireless sensor node [4]. In a model of a wireless sensor node, the time running a program is given by $T = T_{\text{compute}} + T_{\text{communicate}}$, and $T_{\text{communicate}} = N_c (T_s + I_c T_b)$, where $T_s$ is the start-up cost, $T_b$ is the time per byte, $I_c$ is the message length, and $N_c$ is the number of communications [4]. In this section, relationships between power consumption and sensor access modes are discussed in detail. So it is necessary to further divide computation into the process module and sensors. Based on the abstract model in Fig.3, it is obvious that the total power consumption of a wireless sensor node is the sum of power consumption from the three parts, RF module, the process module, and all sensors.

The attributes presented in the model are defined as follows:

- $P_{\text{prod}}$: Power of the process module when the microprocessor is working.
- $P_{\text{pros}}$: Power of the process module when the microprocessor is in a sleep state.
• $T_c(i)$: The time from start to completion of data collection for the $i^{th}$ sensor.
• $T_p(i)$ : The time transmitting the result to the process module from the $i^{th}$ sensor.
• $P_s(i)$: Power of the $i^{th}$ sensor when it is converting or transmitting data.
• $T_{pm}$: Time when the process module is working.
• $W_{pm}$: Total power consumption of the process module, including in the active mode and in the sleep mode.
• $W_{ss}$: Total power consumption of all of sensors during a sampling cycle.

Three assumptions are listed as below:

• A sensor stops work when the data transmission finishes because the system has stopped power supply to the sensor. In other words, there is no power consumption on the sensor.
• Process module only switches between the active state and the sleep state, and there is a large difference of power consumption between two states.
• Power of a sensor that works in converting process is the same as the one in transmitting process.

There could be four sensor access modes to be used for exchanging information between the process module and all sensors. Serial access mode is of that the process module enquiries each sensor in turn; parallel convert mode is of that all sensors sample data synchronically and transmit serially; event-driven mode is of that a sensor informs the process module to receive data when special conditions are satisfied; hybrid mode extracts some merits of three modes mentioned above, but it depends on a real application. Here, only three modes are discussed, and hybrid mode is put into the future research.

A. Scheme 1 - Serial access mode

The process module enquiries all sensors in the system in turn, and the priority of accessing each sensor can be adjusted by programme. After sending a start command, the process module waits for a special period during which the sensor samples and outputs data, and then transmits the results. The mechanism is adapted by the BOS provided by Jennic. According to the assumption mentioned above, it is easy to understand that the time when the $i^{th}$ sensor works is divided as $T_c(i)$, converting time from starting conversion to the data collection being completed, and $T_p(i)$ , transmitting time between the sensor and the process module. So, the total time $T_{pm}$, spent by the process module for processing all sensors, is expressed as Formula (1).

$$T_{pm} = \sum_{i=1}^{n} (T_c(i) + T_p(i)).$$  \hspace{1cm} (1)

$W_{pm}$, the total power consumption of the process module collecting all sensors once, is calculated with Formula (2).

$$W_{pm} = T_{pm}P_{pm}.$$  \hspace{1cm} (2)

$W_{ss}$, the total power consumption of all the sensors, is expressed as Formula (3).

$$W_{ss} = \sum_{i=1}^{n} P_s(i)(T_c(i) + T_p(i)).$$  \hspace{1cm} (3)

B. Scheme 2 - Parallel convert mode

In order to reduce time when the process module waits for the conversion of each sensor, another mechanism is discussed here. After sending a starting command to all sensors and starting the loop of collecting data from sensors, the process module checks continuously the feedback queue that indicates if some of sensors have completed their conversions or not. When the feedback queue is nonempty, the process module starts a transmission from the sensor that completes the conversion first. So the order of accessing sensors depends on the arrangement of sensors in the feedback queue and follows Fist-In-First-Out rule. In other words, a sensor with short conversion time is prior to be accessed. The lower bound of time when the process module works is expressed as Formula (4).

$$T_{pm} \geq \min(T_c(u) \mid u \in \{1,..,n\}) + \sum_{j=1}^{n} T_p(j).$$  \hspace{1cm} (4)

Proof: Assume the conversion for each sensor starts at the time $T_0$ (here $T_0$ is usually assumed as 0), $T_{end}(i)$ is defined as the time at which the $i^{th}$ sensor finishes a transmission with the process module. For the first sensor in the queue, $T_{end}(1) = T_c(1) + T_p(1)$, see Fig. 4. After finishing the transmission with the $i^{th}$ sensor, the process module starts to transmit with the $(i+1)^{th}$ sensor if the feedback queue is nonempty. If the $(i+1)^{th}$ sensor has not yet finished the conversion, i.e. the feedback queue is empty, and the process module has to wait for a gap (see Fig. 5).

According to the assumptions mentioned before, $\forall i, j \in \{1,..,n\}$, $T_{end}(i + 1)$ is expressed as follows:

$$T_{end}(i + 1) = \left\{ \begin{array}{ll}
T_{end}(i) + T_p(i + 1), & T_{end}(i) \geq T_c(i + 1) \\
T_c(i + 1) + T_p(i + 1), & T_{end}(i) < T_c(i + 1).
\end{array} \right.$$  \hspace{1cm} (5)

According to $T_c(i) \leq T_p(i)$, $\forall i, j \in \{1,..,n\}$, $T_{pm}$ is expressed as follows:

$$T_{pm} = \sum_{i=1}^{n} T_p(i).$$  \hspace{1cm} (6)

According to $T_c(i) \leq T_p(i)$, $\forall i, j \in \{1,..,n\}$, and $T_c(i) = \min(T_c(u) \mid u \in \{1,..,n\})$.  

The upper bound of time when the process module works is expressed as Formula (5).
\[ T_{pm}(u) \leq \text{max}(T_s(u) \mid u \in \{1, \ldots, n\}) + \sum_{i=1}^{n} T_p(i). \] \text{(5)}

The total power consumption of the process module collecting data from all sensors once, is expressed as Formulas (6) and (7).
\[ W_{pm} \geq P_{pm}(\text{min}(T_s(u) \mid u \in \{1, \ldots, n\}) + \sum_{i=1}^{n} T_p(i)). \] \text{(6)}
\[ W_{pm} \leq P_{pm}(\text{max}(T_s(u) \mid u \in \{1, \ldots, n\}) + \sum_{i=1}^{n} T_p(i)). \] \text{(7)}

It is known that the process module starts to transmit the data of the \(i\)th sensor after finishing the transmission of \((i-1)\)th sensor. Thus, the shortest time, from \(t_0\) to the time when the \(i\)th sensor completes a transmission with the process module, is expressed as below:
\[ T_{end}(i-1) + T_s(i) \geq \text{min}(T_s(u) \mid u \in \{1, \ldots, n\}) + \sum_{j=1}^{n} T_p(j). \]

The total power consumption of the all sensors for one traversal, is expressed as Formulas (8) and (9).
\[ W_{sa} \geq \sum_{i=1}^{n} P_s(i)(\text{min}(T_s(u) \mid u \in \{1, \ldots, n\}) + \sum_{j=1}^{n} T_p(j)). \] \text{(8)}
\[ W_{sa} \leq \sum_{i=1}^{n} P_s(i)(\text{max}(T_s(u) \mid u \in \{1, \ldots, n\}) + \sum_{j=1}^{n} T_p(j)). \] \text{(9)}

C. Scheme 3 - Event driven mode

Some micro operation systems use an event-driven mode to solve transmissions between sensors and the process module [2], i.e. a sensor requests a transmission with the process module by producing an interruption. The mode can be applied to two cases. One is to process a time-uncertainty emergency sensed by a sensor, the other is for some sensors that their sampling period is too long to be waited by the process module, for example, an electrochemical carbon monoxide sensor mentioned in Section III. The event-driven mode can greatly reduce the power consumption of the process module, but it increases that of sensors because they must run continuously. However, it is a good option, as the power consumption of the process module is usually much larger than the power sum of all these sensors. In order to build a mathematical model for processing a time-uncertainty emergency, a sampling cycle for accessing all sensors is assumed as \(T\), and the frequency of the events occurring during \(T\) is defined as \(f_s(i)\) for the \(i\)th sensor.

There is a comparator inside the sensor circuit, which compares the sampling data with a special threshold. The power consumption of the process module is described as Formula (10).
\[ W_{pm} = P_{pm}(\sum_{i=1}^{n} T_s(i) f_s(i)) + P_{pm}(T - \sum_{i=1}^{n} T_s(i) f_s(i)). \] \text{(10)}

The power consumption of all sensors is calculated with Formula (11).
\[ W_{sa} = T \sum_{i=1}^{n} P_s(i). \] \text{(11)}

For the second case, all sensors of a wireless sensor node have been working during \(T\) and send an interrupt request to the process module while a conversion is completed. The frequency \(f_s(i)\) of transmissions of \(i\)th sensor during \(T\) is expressed as Formula (12). If \(f_s(i)\) in Formula (10) is replaced with Formula (12), the power consumption of the process module is described as Formula (13). The power consumption of all sensors can be calculated with Formula (11).
\[ f_s(i) = \frac{T}{T_s(i) + T_p(i)}. \] \text{(12)}
\[ W_{pm} = P_{pm} T + (P_{pm} - P_{pm5}) \sum_{i=1}^{n} \frac{T_s(i) + T_p(i)}{T_s(i) + T_p(i)}. \] \text{(13)}

Define \(\alpha(i) = \frac{T_p(i)}{T_s(i) + T_p(i)}\), where \(\alpha(i)\) is only related to the attributes of the \(i\)th sensor, then Formula (13) becomes:
\[ W_{pm} = T(P_{pm1} + (P_{pm1} - P_{pm}) \sum_{i=1}^{n} \alpha(i)). \] (14)

V. EVALUATION

Once the selection of all components of a sensor node is completed, many parameters that are defined above will be confirmed. According to the mathematic models described in Section IV, the power consumption of a sensor node can be calculated. Because there are many factors to affect the power consumption of a system, such as the run-time overlapping among subsystems, the differences of power consumption among electronic components, the time differences of conversion and transmission for heterogeneous sensors, it is considerable to evaluate the mathematic models corresponding to different sensor access modes in order to select optimal sensor access modes with energy awareness.

Scheme 1 can access all sensors in a presetable order, and it is possible to assign some priorities for different sensors. It is obvious that Scheme 2 has an unchangeable order to access all sensors, as the sensor with smaller conversion time has high priority. Scheme 1 makes the process module have a longer running time to access all sensors than Scheme 2 does, as the difference by subtracting Formula (5) from Formula (1),

\[ \sum_{i=1}^{n} T_s(i) - \max(T_s(u) \mid u \in \{1, \ldots, n\}) \] is positive. In general, \( P_{pm1} \) is much larger than the power sum of all sensors, so the power consumption in Scheme 1 is larger than that in Scheme 2 for the process module. For sensors, the power consumption in Scheme 2 is larger than that in Scheme 1 according to the comparison between Formula (3) and Formula (7). It is difficult to compare the total power consumption of a sensor node for the two schemes, unless all parameters of a sensor node are confirmed in a real application.

Scheme 3 can significantly reduce the power consumption of the process module, so many of micro operation systems have adopted it as one of the work modes for accessing sensors. It is difficult to calculate the exact result of Formula (10), because the sampling cycle \( T \) for accessing all sensors and the frequency \( f_s \) of the events occurring during \( T \) depend on a real application, and \( f_s \) is random, even though all parameters of a sensor node have been evaluated. Formula (13) also depends on a real application for parameter \( T \). At least it should be guaranteed that the process module can access all sensors during \( T \). But the mathematic models provided by Scheme 3 are very useful to build a simulation of a wireless sensor node designed for a special application.

VI. CONCLUSION AND FUTURE WORK

There are very large differences of power consumption of sensor nodes while the different sensor access modes are applied, even though they are composed of the same electronic components and implement the same goal. The mathematic models based on three sensor access modes in this paper provide a basic prototype to simulate a task scheduling system with energy awareness. The paper also provides valued results to designers who are engaged in the implementation of a micro operation system (MOS) applied for embedded systems so that the MOS can provide a flexible framework to support different sensor access modes. It is very important to build a simulation system for a wireless sensor node in the future work to evaluate a design scheme with energy awareness.

REFERENCES