

Energy Characterization of End-Device in ZigBee-based Wireless Sensor Network

Yikui Zhang¹, Xiangbin Xin¹ and Youmeng Li^{1,a}

¹ School of Computer Software, Tianjin University, Yaguan Road 135#, Jinnan District, Tianjin 300350, China

Abstract. In this paper, a flow of ZigBee end-device energy characterization is introduced. The variation of the energy and power consumption of wireless communication through ZigBee end-device is studied. The remainder of this article details the methods used to determine energy and power overheads of the end-device in four parts: start-up stage, basic energy expenditure, data-sending consumption and polling consumption. The impacts of software parameters like polling rate and data size on the energy consumption are analysed. Also, models of the power and energy are extracted. Our methods allow estimating the energy consumption of the ZigBee end-device when running a Z-Stack application on a specific hardware platform.

1 Introduction

ZigBee Wireless Technology is emerging as the leading method for implementing low-cost, low-data rate, short-range wireless networks with extended battery life for use in creating the “Internet of Things” [1]. In recent years, ZigBee wireless sensor networks are becoming widely applied in a large number of fields including health, agriculture, industrial and geology, military, home and emergency management. Most of these networks are composed of multiple sensor nodes, which are able to collect specific data like temperature, humidity, etc.

In ZigBee networks, there are three kinds of nodes: coordinator, router and end-device. End-Device can connect to coordinator and router but cannot support routing [2]. It is a ZigBee end-device that joins a network and normally turns off its receiver when idle. This requires its parent to hold messages until the end-device polls for its messages. In most cases, an end-device is equipped with some sensors for collecting data and is powered by batteries. So energy consumption of end-device is a vital aspect for software and hardware designers to consider. Accurate energy characterization can make it easy to estimate battery life according to its capacity, or to choose optimum battery capacity according to designed life cycle. In our work, we proposed a method and flow of characterizing energy and power consumption of end-device in ZigBee network, taking both hardware factor and software factor into consideration.

2 Related works

In order to characterize energy consumption and establish energy models on embedded system, several studies have proposed their approaches.

Dhouib et al. [3] presented a multi-layer power and energy estimation approach for embedded applications and operating systems. According to them, energy can be summed over the hyper period of the real-time system. They estimated energy and power consumption of standalone tasks and also added energy overheads of embedded operating system services such as timer interrupt, IPC and peripheral devices accesses.

Xia et al. [4] propose a quantitative approach, in which EOS and applications are executed on two cooperated simulators, to estimate and analyse the energy consumption of embedded operating system (EOS). They used a cycle-accurate micro-architecture power model to estimate instruction execution energy.

Waseem [5] discussed and compared currently used energy efficient mobile operating system. And then he summarized and discussed recent energy efficient techniques used to reduce the power consumption of mobile devices.

Baynes et al. [6] described their simulation environment and the modelling of embedded systems with SimBed, which measures the execution behaviour and power consumption of the RTOS and embedded applications. The authors compared three different RTOSs: μ COS, Echidna and NOS. However, their models are not deduced from measurements on actual hardware platform.

3 Energy characterization and estimation flow

Basically, the energy consumed by an end-device in a certain time duration T after it's powered up can be divided into four parts: start-up stage, basic energy expenditure, consumption and polling consumption.

^a Corresponding author: liyoumeng@tju.edu.cn

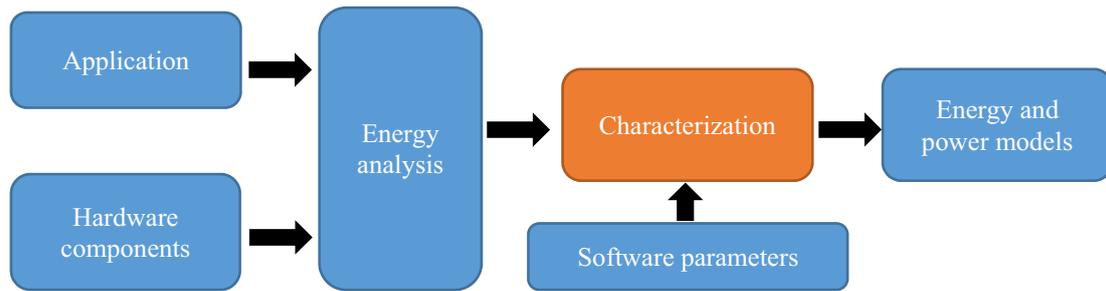


Figure 1. The methodology of end-device energy characterization.

Total energy can be given by Equation (1):

$$E_T = E_0 + E_b(T) + E_d(T) + E_p(T) \quad (1)$$

The proposed method consists in extracting an energy model of the end-device power overhead. The inputs are the generic application and the hardware platform. As showed in Figure 1, to characterize the energy consumed by the end-device, it's necessary to analyse the application running on the board. In the energy analysis step, a set of software parameters, which influence the energy consumption, are varied and then energy profiles are traced. The energy traces obtained are able to characterize the energy overhead of the end-device and then to model the power and energy consumption. After extracting the energy models, we determine the energy and power laws, as showed in Figure 2.

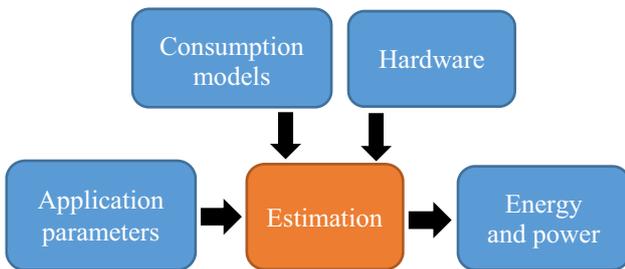


Figure 2. Estimation of the end-device energy consumption.

In Equation (1), T represents the total working time of the end-device. E_0 represents the energy consumption in the start-up stage, which is from being powered to successfully adding to the ZigBee network. In this stage, an end-device has to finish a few necessary activities, including hardware initialization, OS abstract layer initialization, ZigBee network discovery, network-adding request, and acquiring IP address. For a certain hardware platform, completing this stage takes a proximately constant time T_0 . So T has no effect on E_0 when T is bigger than T_0 and we can make a presumption of $T \gg T_0$.

$E_b(T)$ represents the basic energy consumption of the end-device from time T_0 to T and it's the energy consumed by core circuit and peripheral circuit to maintain basic operations except sending data. To characterize the basic energy consumption, we need to stop the end-device from sending data and polling messages.

$E_d(T)$ represents the extra energy for sending data to the ZigBee coordinator during the time from T_0 to T. An

end-device is usually required to send fixed size of data to the coordinator periodically.

This energy is divided into two parts: data transmission and state switch.

$E_p(T)$ represents the energy for polling from time T_0 to T. End-Devices in the Z-Stack sample applications are setup initially with automatic polling for messages enabled. Three different polling options are supported, each controlled by a time delay parameter. If the end-device never needs to receive messages once it has joined the network, all three polling rates should be set to zero.

To characterize the energy overhead of the end-device, we should analyse the four parts and extract their energy models separately. In the next section, we build models for the four parts with the help of current and voltage data samples acquired by power monitor.

4 End-device power and energy models

4.1 Start-up stage

The trace of electric current change in start-up is showed in Figure 3(a). The last peak in Figure 3(a) indicates that the end-device has successfully joint into the ZigBee network and acquired its IP address. We sampled the data 10 times and calculated the energy for each, shown in Figure 4. As the calculated numbers has slight fluctuations, we consider E_0 a constant value and define it as the mean value of these samples.

Finally, E_0 equals 671.0121 mW*s and T_0 equals 3.27s.

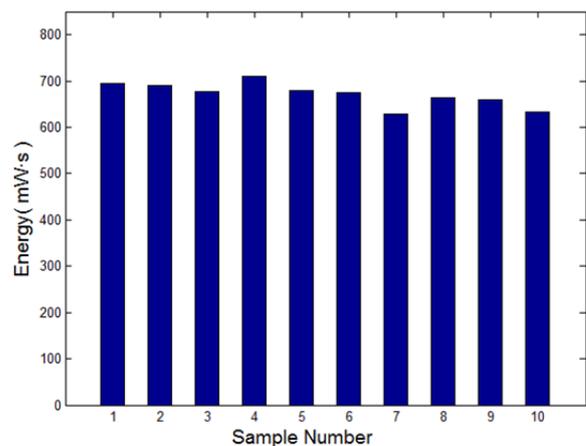


Figure 4. Ten energy samples in start-up stage.

4.2 Basic energy consumption

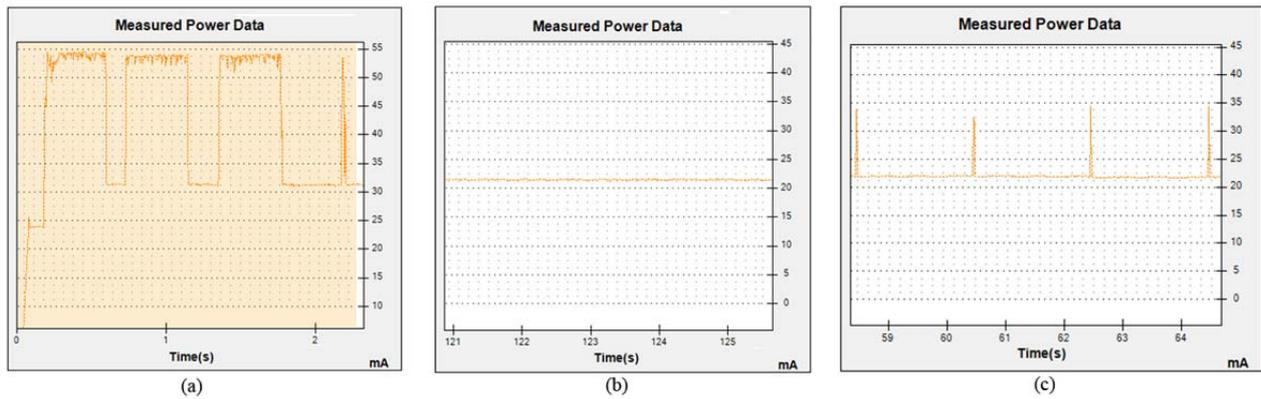


Figure 3. Traces of measured real-time current.

As is shown in Figure 3(b), when an end-device gets a steady state without data transmission and polling, its current-changing trace is similar to a horizontal line. So we define $E_b(T)$ by the Equation (2):

$$E_b(T) = P_b (T - T_0) \quad (2)$$

In Equation (2), P_b represents basic power and it's also a constant value and calculated following the Equation (3):

$$P_b = \sum_{i=1}^N U_i I_i / N \quad (3)$$

Where N is the sample size, U_i and I_i are respectively the voltage and current value of a sample point. We obtained about 25 thousand sample points within 5 seconds. Finally, P_b equals 106.8335 mW.

4.3 Data-sending energy consumption

Mostly, an end-device needs to report data to coordinator regularly when it's working. Our work here is to find out the relation between energy consumption and data size for one sending activity. As showed in Figure 3(c), each peak represents a sending activity and the sending process is extremely short. Energy consumed in this process can be calculated through integration. As this energy is consumed in two ways: changing device state and emitting electromagnetic waves through an antenna. The first part is stable and the second part is varied by the data size. We can define the Equation (4):

$$e_d = kS + e_0 \quad (4)$$

Where e_0 represents the energy consumed for changing device state, S represents the data size and k is the coefficient.

Figure 5 depicts the variation of measured data-sending energy consumption with data size. According to the fitting curve, k equals 0.0110 and e_0 equals 0.8573.

Meantime, $E_d(T)$ is varied by both S and time interval T_{si} , as showed in Equation (5):

$$E_d(T) = (kS + e_0) * (T - T_0) / T_{si} \quad (5)$$

4.4 Polling energy consumption

If an end-device needs to receive messages from other nodes, polling-rate must be set non-zero. The energy consumed for polling during a period of time is only varied by polling time interval T_{pi} . So $E_p(T)$ can be calculated by Equation (6):

$$E_p(T) = \begin{cases} e_p * \frac{T-T_0}{T_{pi}}, & T_{pi} \neq 0 \\ 0, & T_{pi} = 0 \end{cases} \quad (6)$$

Where e_p represents the energy consumed for polling once. e_p can be calculated through integration and it is assigned with the mean value of 100 sample, equaling 0.7032 mW*s.

From all the above, the total energy consumed by an end-device can be estimated when the following four parameters are given: working time T , data size S , sending time interval T_{si} and polling interval T_{pi} .

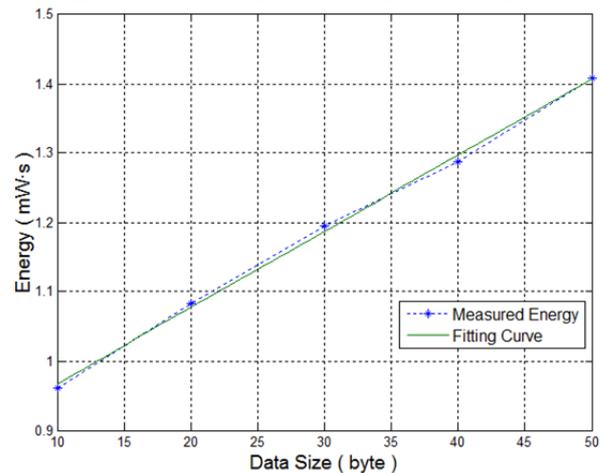


Figure 5. Energy variation as a function of data size.

5 Experiment

5.1 Hardware platform

The target hardware platform is the CC2530EB board from TI (Texas Instruments) [7]. It is equipped with an industry-standard enhanced 8051 processor as its MCU (Micro Controller Unit) [8]. The environment we build application on is ZigBee protocol stack (Z-Stack) from Texas Instruments. IEEE standard 802.15.4 defines physical (PHY) layer and media access control (MAC) layer [9]. Z-Stack Home is TI's ZigBee Home Automation (ZHA) compliant protocol stack for the CC2630, CC2530 and CC2538 System-on-Chip [10]. As shown is Figure 6, We use Mobile Device Power Monitor hardware and Power Tool software form Monsoon Solutions, Inc. to measure the power dissipation of an end-device. The Power Monitor is connected to a computer through a USB cable and an end-device is powered by the Power Monitor through a USB cable. The raw data file we get from Power Monitor is in the format of CSV, containing large amount of number pairs -- count time (s) and electric current (mA). These discrete data points can be processed with Matlab software through trapezoid integration to calculate the gross of energy consumed.



Figure 6. The Measurement Framework.

5.2 Experiment design and results

To verify the energy and power model, four experiments with variation of four parameters are performed on the hardware platform that is introduced above. With the Power Monitor successfully connected to the computer and IAR embedded workbench installed and open, we started the experiments. Initially, we set the polling off and make the end-device send 15 bytes data to the coordinator, which had already setup a ZigBee network, every each second. After 20 minutes, we turned off the end-device and export raw data file in the format of CSV. Then the file is simply processed by deleting its first title line so that it can be read by Matlab. Finally, the total energy consumed was calculated through integration. The application were compiled and loaded repeatedly with varied parameter combination. These groups of parameters and experimental results are shown in Table 1.

Table 1. Experimental results

No.	T (s)	T _{pi} (s)	T _{si} (s)	S (byte)	Estimated (mW·s)	Measured (mW·s)	Error (%)
1	1200	0	1	15	129745.2836	130203.6488	0.35
2	1200	1	2	25	130040.9358	130354.7847	0.24
3	1200	2	3	35	129438.2027	129023.0945	0.32
4	1200	3	4	45	129206.9646	129432.4539	0.17

According to Table 1, we note that the average estimation error is 0.27%. Actually, this statistics can't strongly prove the correctness of all our models. Because the basic energy consumption accounts for a huge proportion of total number. So we have to exclude basic energy to validate the other energy models, which is shown in Table 2.

we excluded basic energy and calculated the overall energy of the other three parts, because their values are in the same order of magnitude. Finally, according to Table 2, the average estimation error equals 3.20%, which proves that our models work well in this situation.

Table 2. Experimental results without basic energy

No.	T (s)	T _{pi} (s)	T _{si} (s)	S (byte)	Estimated (mW·s)	Measured (mW·s)	Error (%)
1	1200	0	1	15	1894.4292	1987.0564	4.67
2	1200	1	2	25	2190.0813	2135.1457	2.51
3	1200	2	3	35	1587.3483	1622.1621	2.19
4	1200	3	4	45	1356.1101	1402.6938	3.44

5.3 Deviation analysis

In our experiments, there are two main deviation sources: the Power Monitor and end-device itself. The Power Monitor we use is able to measure the current accurately, with a dual range, self-calibrating and integrating system used [11]. But it may still drop samples when running. The end-device may lose packets during the transfer process, especially when there exist WIFI or Bluetooth networks in the lab environment, which would surely increase the energy consumption as a result.

6 Conclusion

We have divided energy consumption of an end-device into four parts and proposed energy and power models of each part. The models are based on measurements on the hardware platform CC2530EB board and allow the characterization of the energy overhead of the end-device. Experiments show that these models can precisely characterize the energy overhead. For this reason, we plan, in the future work, to analyse more details of data-sending process and basic energy consumption. Apparently, these future works of this project will focus on optimizing the energy consumption of the end-device.

Acknowledgements

Our work was supported by Tianjin University-Qinghai Nationalities University Special Independent Fund, with the project number: 2015XZC-0038.

References

1. YF Zhong, YJ Liu. ZigBee wireless sensor networks [M]. BeUing University of posts and telecommunications press, 2011.
2. Abhishek Kumar, Sandeep Gupta, "Study on ZigBee Technology", IEEE, 3rd International Conference on

- Electronics Computer Technology (ICECT), Volume **6**, Pages: 297 – 301, 8-10 April 2011.
3. Saadia Dhouib, Eric Senn, Jean-Philippe Diguët and Johann Laurent, Modelling and Estimating the Energy Consumption of Embedded Applications and Operating Systems, ISIC-2009 - 12th International Symposium on Integrated Circuits, Proceedings, p 457-461, 2009.
 4. Zhao Xia, Guo Yao, Lei Zhi-Yong and Chen Xiang-qun, Estimation and analysis of embedded operating system energy consumption, Acta Electronica Sinica, v **36**, n 2, p 209-15, Feb. 2007.
 5. Muhammad Waseem, Energy efficient mobile operating systems, International Journal of Advanced Networking and Applications, v **5**, n 1, 1812-17, July-Aug. 2013.
 6. K Baynes, C Collins, E Fiterman, B Ganesh, P Kohout, C Smit, T Zhang, BL Jacob, The performance and energy consumption of embedded real-time operating systems. IEEE Trans. Comput. **52**, pp. 1454–1469 (2003)
 7. Texas Instruments. CC2530 Datasheet [EB/OL]. <http://www.ti.com.cn/cn/lit/ds/symlink/cc2530.pdf>
 8. Second Generation System-on-Chip Solution for 2.4 GHz IEEE 802.15.4 / RF4CE / ZigBee(2015), <http://www.ti.com/product/CC2530/description>
 9. ZigBee Alliance. “ZigBee Overview”. September 2003.
 10. ZigBee protocol stack, <http://www.ti.com.cn/tool/cn/z-stack#descriptionArea>
 11. Power Monitor End User Manual, [http://msoon.github.io/powermonitor/PowerTool/doc/Power Monitor Manual.pdf](http://msoon.github.io/powermonitor/PowerTool/doc/Power%20Monitor%20Manual.pdf)