Energy Measurements for MicaZ Node

Marc Krämer, Alexander Geraldy

University of Kaiserslautern, Germany
{kraemer,geraldy}@informatik.uni-kl.de

Abstract. Energy consumption and power supply are important aspects of mobile devices. To determine the operation time, knowledge about the energy consumption of the mobile device and the capacity of the battery are needed. By using energy saving techniques and graceful degradation, we can extend the life time significantly. To use these techniques efficiently with respect to response time and functions of the device, measurements on the device and the battery are required.

1 Introduction

Mobile devices are often powered by batteries. Manufacturer of rechargeable batteries specify the capacity of the cell, which is determined using measurements on a certain load. For lithium polymer batteries this load is often relatively high compared to the consumption of a small device. The usable capacity of a battery depends mainly on the minimum operational voltage needed by a device, the current and the temperature. To determine the runtime of a mobile device, there are two things necessary: the useable capacity of the battery and the energy consumption of the mobile device. Since the device can be in idle state, computing, transmitting or in some energy saving state, its energy consumption varies.

If the device is never idle, it is hard to save energy by using the standard saving techniques, i.e. switching to energy saving mode in idle period. Even if these modes should be used, the problem is that the microcontroller of the MicaZ provides six energy save modes to choose from. We propose a graceful degradation of functionality if the battery is running low. Graceful degradation is the degradation of a system in such a manner that it continues to operate, but provides a reduced level of service rather than failing. For example consider a device with some core functions and additional functions, that works for some time t. If the battery is getting low, the additional functions are degraded, which means they are not switched off abruptly, but their runtime is shortened and finally set to zero. Even the core functions may have parts that can still work with less runtime or the execution frequency can be degraded. During the new idle periods it is now possible to switch to sleep modes and save energy. To determine when the battery is at a low level, measurements for the used battery are required.

Switching to save modes has some side-effects, i.e. stopping the CPU, stopping of timers or discarding external events. Therefore the requirements of the application running on the device need to be considered as well as the consumed energy.

2 Related Work

For the Mica2 node, an earlier version of sensor nodes from Crossbow, there exist a few measurements [1] and an integration of these measurements into AVRORA [2], a simulator for ATMEL AVR microcontrollers. Since the Mica2 node uses a different transceiver than the MicaZ node, at least measurements of the new transceiver are required. Since the redesign of the node for the new transceiver can have influence on the energy consumption, the other components have to be measured too.

Lithium polymer batteries are often used for high current purposes, so there exist measurements with high loads (some amperes). We have only low current (a few mili-amperes), so we cannot rely on the specifications of the manufacturer of the batteries.

3 Measurements of a Lithium Polymer (LiPo) Battery

For most applications, it is necessary to know when the battery capacity is below a certain level. If the battery is below that level, a warning can be given or a controlled shutdown of the device can be made. We chose the LiPo batteries, because mobile devices should have low weight energy source with high energy density. For the used LiPo batteries there exist only a few discharging curves and most of them use a discharge current at a level multiple of the capacity [3]. The reason for this is, that they are often used in model aircrafts, where high current is needed. In our case, where no appropriate curve is available, we have to make the measurements on our own.

3.1 Experiment Setup

For the experiment, we use a lithium polymer rechargeable battery from Kokam [4] with a nominal capacity of 1.5 Ah and a nominal voltage of 3.7 V. We did the discharge at a constant current of 37.8 mA which is ensured by an added DC/DC regulator and a constant resistor. The voltage of the battery, as well as the output voltage of the DC/DC regulator are recorded by a digital oscilloscope [5].

3.2 Experiment Results

The result of the experiment is shown in Figure 1. On the x-axis, the time is in units of 30 seconds, on the y-axis the voltage of the battery and the output of the DC/DC regulator is drawn. The curve can be divided in four parts. The first section, till 250 time units, is decreasing slightly faster than the middle section. In both sections, the relationship between voltage and time is nearly linear. Only in the last section, at about 3250 time units, the voltage is roughly constant for a small amount of time and afterwards decreasing really fast to the shutdown point of the voltage regulator at $t = 3816$.

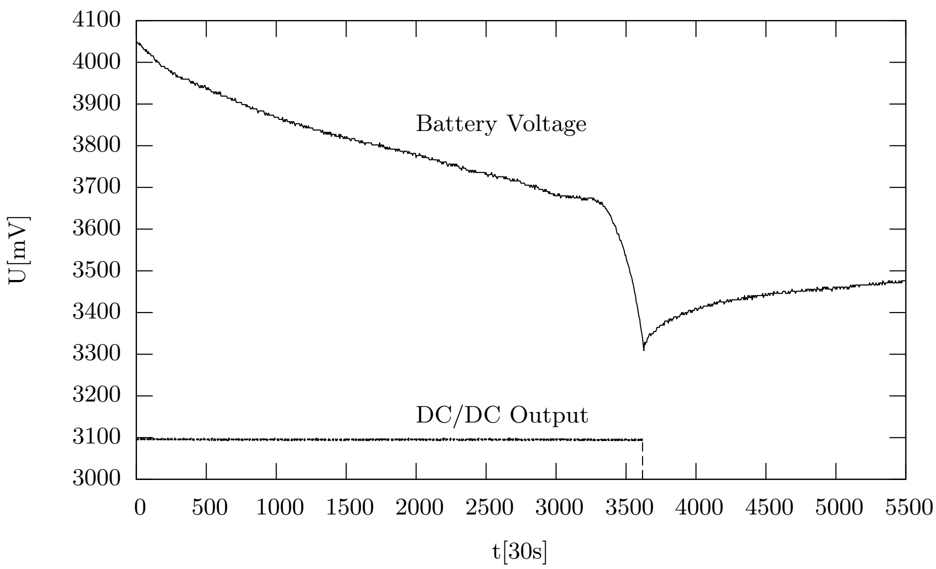

Fig. 1. Battery discharge at 37.8 mA

The useable capacity can be calculated to $\frac{3816 \cdot 30\mathrm{s}}{3600\frac{\mathrm{s}}{\mathrm{h}}} \cdot 37.8\,\mathrm{mA} = 1202\,\mathrm{mAh}$. The important result of this experiment are the voltage changes at discharge. Due to the correlation between current battery voltage and the capacity left in the battery it is sufficient to measure the current voltage. This fact is interesting, because the voltage can be determined by the MicaZ using its built-in AD converter.

4 Measurements of the MicaZ Node

To determine the operating time of the MicaZ node [6], exact measurements on its energy consumption are needed. The MicaZ node consists of a microcontroller [7], a transceiver (transmitter/receiver) [8], three light emitting diodes (LED) and a few other components that are not relevant here. All components can be switched on, off or have some energy saving mechanisms. For most of these electrical components, we can find the energy consumption in their data sheets. The sum of all these ratings may differ from the real value, due to additional components like resistors etc. Another problem is that for some functions, e.g. sleep modes, no power consumption is given. In the master's thesis of Dominik Domis [9], measurements of another node called *particle* [10] show that there can be a big gap between the data presented in the data sheet and the measured value. According to this, measurements of the microcontroller in different operation modes, the transceiver and the LEDs were performed. Since the node is considered as a whole inseparable node, the measurements were taken with the whole node. To get better results, each unused component in a specific measurement is put into its highest energy save mode.

4.1 Experiment Setup

For this experiment, we use the same DC/DC regulator as in the prior experiment to reduce the voltage to 3.1 V for the MicaZ. At the output of the regulator we add the MicaZ. For the measurement, the current of the MicaZ and the regulator as well as the voltage of the battery are recorded by a digital oscilloscope [5].

4.2 Experiment Results

For each of these states, we did measurements of the energy consumption of the MicaZ. Since the MicaZ needs to be considered as a whole, the consumption of the node with all components shutdown is the minimum energy consumption of the node, even if no program is running on it. We perform our measurements including the DC/DC regulator, because it is needed for a proper functionality of the node.

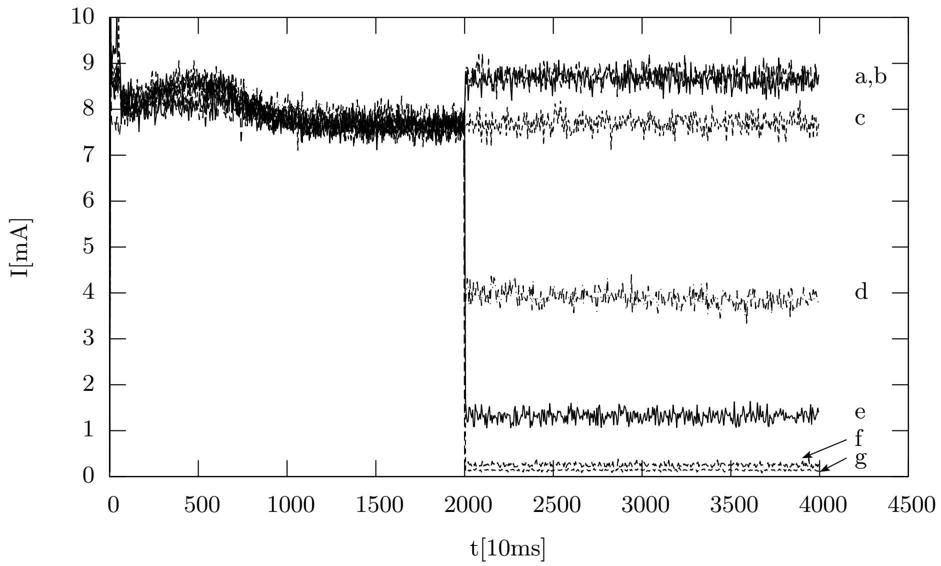

Fig. 2. Energy save modes of ATMega 128L – a:Busy (mul), b:Busy (jmp), c:NOP, d:Idle, e:ADC, f:Ext. Standby, g:Save

Microcontroller

The microcontroller of the MicaZ is an ATMEL ATMega 128L, which has various functions to save energy. The obvious ones are the six different energy saving modes. Another one is to decrease the operation frequency by the built-in frequency scaler. Figure 2 shows the different save modes and their consumption. Before entering any of these modes, the program waits a while to stabilize the consumption, which can be seen in the figure.

The first three measurements concern normal operation mode. Two of them represent duty cycles, while the third one is an idle loop. Both busy loops have a high consumption, while the NOP loop has a lower consumption of 1 mA. All other measurements show energy saving modes. The idle mode has the lowest savings, but has the advantage that almost all parts of the microcontroller still work. All other modes, save more energy, but have side-effects resulting from disabling more parts of the microcontroller. In this measurement, the standby and the powerdown mode have not been considered. These two modes stop the main clock, so a wake up is not possible after a defined time, which is needed for our future work. Only an external interrupt can wake up the device. In Table 1, the average consumption of the microcontroller in every mode is listed. Details on different modes are given in the data sheet [7, p. 44].

Operation Mode	average current [mA]
Busy (mul)	8.65
Busy (jmp)	8.73
NOP	7.69
Idle	3.88
ADC	1.32
Extended Standby	0.25
Save	0.14

Speed Level	average current [mA]
1/2	4.96
1/4	3.57
1/8	2.73
1/16	1.65
1/127	0.76
1/2 + Idle mode	2.23
1/16 + Idle mode	0.84

Table 1. Average energy consumption

The second way for saving energy is decreasing the internal operation frequency. If the next task should be executed at time t, the frequency can be decreased to a level where the current task ends at time t. Therefore no waiting or switch to a energy save mode is needed. An example for this is given in Figure 3. By measurement we determined that the consumption at $\frac{\text{speed}}{32}$ is not $\frac{P}{32}$, but it is $\sim \frac{1}{8}P$. Considering this, the expected consumption in example in Figure 3 can be calculated to 13.8 J. So there is no energy saving anymore. Nevertheless, if we cannot use a sleep mode, this technique can have some benefit. Consider the CPU is waiting for some external event by using a busy loop, or timers are running, thus entering a *deep sleep mode* is not possible. In many sleep modes the CPU or the timers except timer 0 are stopped, so entering any sleep mode is impossible. In this case, one can scale down the frequency i.e. by 16 and enter the idle mode. Since we have nothing to compute and are only waiting for an event, the energy consumption for the same time of waiting is lower, without entering a save mode. By comparing both values in Table 1 one can see, that the consumption at Speed/16 in combination with idle mode is even lower than in ADC mode.

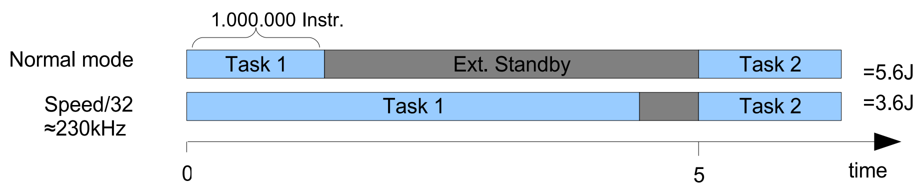

Fig. 3. Theoretical benefit of decreasing frequency on ATMega 128L

Transceiver
The transceiver is a component which has the highest energy consumption of all relevant components of the MicaZ. This component has four different states: *down*, *idle*, *send* and *receive*. According to the data sheet, the receive mode should have a higher consumption than the send mode. By measurement we determined the current in both modes to 20 mA. The idle mode has a lower power up delay and the consumption is about 0.4 mA higher than in down mode.

Figure 4 shows the mentioned measurements. For this measurement, the current of the microcontroller is not interesting and should be low to get higher precision. Therefore the frequency is set to $\frac{1}{16}$. The transceiver initializes and is set to receive mode. After this period a send operations is started, which can not be seen by measurement and finally the transceiver is automatically set back to receive mode. After the measurement the transceiver is set to down state again.

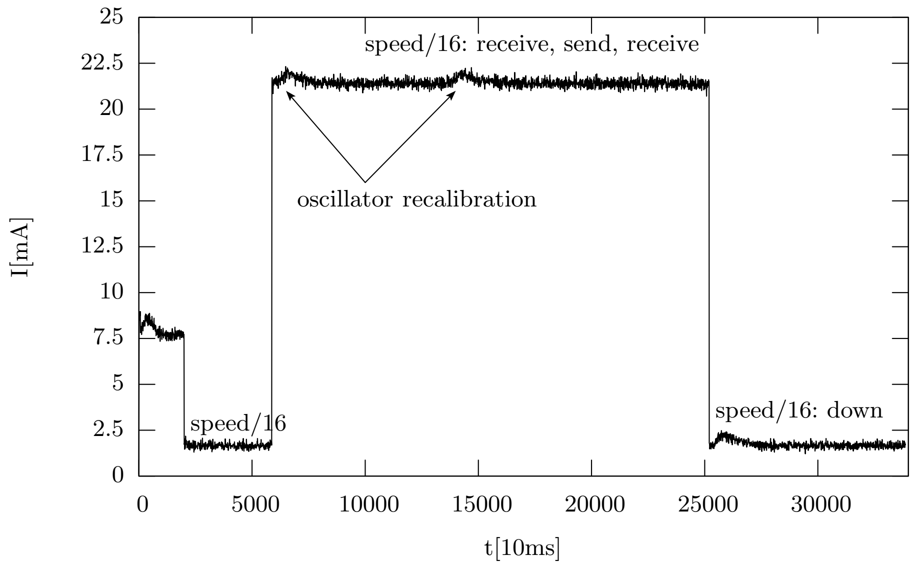

Fig. 4. Energy consumption of transceiver CC2420

LEDs
For each of the LEDs we determined an average consumption of 10.2 mA including the processor. By subtracting the NOP consumption (7.69 mA) of the microcontroller, as stated in Table 1, the consumption of a LED is 2.5 mA.

Other Components
The other components on the MicaZ, like the serial data logger or the unique identifier, are not interesting for our research or only used once at start-up and therefore have not been analyzed.

5 Conclusion

Measurements on LiPo batteries have shown, that there is a good relation between the capacity left in the battery and the current measured voltage. Therefore it is easy to determine the current battery level. Hence it is now possible to achieve a graceful degradation of functionality of the device. With the measurements of the device, we can determine how much energy is consumed in each mode. Some power saving modes that seemed useful while reading the data sheet cannot be used to save energy. However, the promising method of frequency-scaling can be used for power saving whenever the sleep mode is not applicable.

Using these measurement results, we can improve the precision of the energy component of the AVRORA simulator. Thereby we can simulate energy scheduling techniques without measurement. In the next step we will develop a dynamic scheduler, which will on the one hand schedule tasks based on the current energy level, the respective power consumption of the tasks and their priorities to enable graceful degradation. On the other hand, we will be able to decrease the frequency of the microcontroller to execute the tasks in time but with minimized energy consumption.

References

[1] Olaf Landsiedel, Klaus Wehrle, Stefan Götz: Accurate Prediction of Power Consumption in Sensor Networks. In: Proceedings of The Second IEEE Workshop on Embedded Networked Sensors (EmNetS-II), Sydney, Australia (2005)
[2] AVRORA: AVRORA Homepage. http://compilers.cs.ucla.edu/avrora/ (2006)
[3] Rinninsland, D.: Discharge curve of a LiPo battery. http://www.lipoly.de/modellbau/akkus/kokam/kokam2000hd04.gif (2006)
[4] Kokam: Kokam 603870H. http://www.kokam.com/product/product_pdf/high_power/PM-201_SLB603870H_1500mAh_Grade.pdf (2006)
[5] Meilhaus: Meilhaus Electronic DS1M12. http://www.meilhaus.de/e_me/ds1m12.htm (2006)
[6] Crossbow: Crossbow datasheet on MicaZ. http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf (2006)
[7] ATMEL: ATMEL ATMega 128L datasheet. http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf (2006)
[8] Chipcon: Chipcon CC2420 datasheet. http://www.chipcon.com/files/CC2420_Data_Sheet_1_3.pdf (2005)
[9] Domis, D.: Komponentenbasierte Energiemodellierung am Beispiel eines Ambient Intelligence Systems. Master's thesis, Fachbereich Informatik, Technische Universität Kaiserslautern (2006)
[10] Particle Computer: Homepage of Particle Node. http://www.particle-computer.de (2006)