Power Considerations When Using High Capacity Data Storage On Wireless Sensor Motes

Michael Healy, Thomas Newe and Elfed Lewis
Department of Electronic & Computer Engineering
University of Limerick
Ireland
Email: michael.healy@ul.ie

Abstract— **In recent years the onboard storage on wireless sensor motes has grown very dramatically, going from Kilobytes (KB) of available space to Gigabytes (GB). This massive increase has primarily come from the addition of support for micro or mini Secure Digital (SD) flash cards on the nodes. This extra storage capacity has led to new use cases for sensor motes which result in fewer data transmissions as a result of more in network aggregation and processing of the sensor data. The primary motivation for using this approach is that writing data to, and then reading data from the SD card, aggregating and processing this data before transmitting smaller packets, should be much more power efficient than transmitting the raw data using the onboard radio. We investigate the power profiles of applications that use SD cards for this purpose versus those that do not in order to determine if there is in fact any power savings, and if so, exactly how much energy can be saved.**

I. Introduction

Secure Digital (SD) is a format of removable flash memory cards, popular for use in mobile devices such as cellular phones, digital cameras, portable media players, etc. Originally developed by Panasonic, SanDisk Corporation and Toshiba Corporation the format is now controlled and developed by the SD Assosiation, a "global ecosystem" of approximately 1,100 companies [1]. SD cards come in three physical sizes, standard, mini and micro, with the mini format being approximately half the size of the standard card and the micro card approximately half the size of the mini. In terms of capacity, normal SD cards are available up to 4 GB but some hosts still only support cards with a maximum size of 1 GB or 2 GB as they were designed to be compatible with different revisions of the SD specification. Higher capacity cards are available, up to 64 GB but they are SD High Capacity (SDHC) or SD Extended Capacity (SDXC) cards. Hosts are not required to support these extended specifications to be considered SD compatible.

A number of newer generation wireless sensor motes have built in support for SD cards or have add on boards to enable this support. Nodes with this feature include the Intel SHIMMER [2], the Sun Microsystems SPOT [3], the MSB mote from FU Berlin [4], the Fleck mote from CSIRO ICT centre [5], etc. As the price of these SD cards has dropped dramatically in recent years and months, to a couple of dollars for a 2 GB microSD card, they now offer a viable low cost alternative to blindly sending all sensed data over the radio to a base station. Often however, the cost, in terms of power consumed, of making use of this feature can be much higher than expected.

We investigate the power consumption of using these cards on wireless sensor motes versus sending the sending the same data over an 802.15.4 radio [6].

II. Experimental Setup

In order to establish the true cost of making use of this type of external flash memory we ran a number of tests using a SHIMMER mote and a variety of microSD cards.

We tested multiple cards from a number of manufacturers, of varying sizes, and of varying ages. We did this for a number of reasons. Firstly we wanted to determine the consistency of power consumption across a range of microSD cards and whether any variations could be tied to the size of the card or to the manufacturer. We also wanted to test if newer microSD cards performed better than old ones, or did the amount of use the microSD card had seen make a difference. The six microSD cards we tested with are shown in table 1. An important point to note is that the age of the card given in this table is only a rough guide as it gives the time since the card was purchased, not since manufacture. It is very hard to tell how long the card had been sitting in storage before it was purchased. The column showing the amount of use the microSD card has seen is also only a relative indication, the cards shown as having heavy use were used in mobile phones or portable media players for a number of months/years before being used for these tests, whereas the cards with light use were never used for any other function until we performed these tests.

Our test applications are written for TinyOS [7], the predominant operating system for wireless sensor networks. TinyOS supplies the drivers necessary to access the microSD cards on the SHIMMER and control the radio transmission, as well as managing the scheduling and performing power management. The version of TinyOS used was tinyos-2.x checked out from Sourceforge's TinyOS CVS tree [8] on May

TABLE I. SD Cards Used in Tests

Manfacturer	Size	Approx. Age (months)	Amount of use before tests
SanDisk	2 GB	10	Light
A-Data	2 GB	6	Light
SanDisk	1 GB	15	Heavy
A-Data	1 GB	6	Light
SanDisk	64 MB	24	Medium
Kingmax	64 MB	30	Heavy

7th, 2009.

To negate any bearing the starting state of the microSD card would have on our power measurement tests, i.e. the initial data on the card, every single data bit on each card was set to zero before each test.

The power consumption was measured using an Agilent 66321D mobile DC source with battery emulation [9] in conjunction with Agilent 14565B device characterization software [10]. Using these tools the current being drawn by the SHIMMER was constantly monitored, and the average current drawn over a period of time was calculated with a high degree of accuracy.

The main point of these tests is to find the impact of writing data to the microSD card versus sending the same data over the radio has on the battery life of a wireless sensor mote. As battery life is generally given in Ampere-hours (Ah) or milliAmpere-hours (mAh), all of our results are given in terms of the amount of current drawn by the SHIMMER over a period of time in order to allow easy calculation of how long a battery will last with the particular application. For the purposes of our discussions we use the Ultralife UBC005 [11] as our reference battery, a rechargeable lithium ion battery commonly used with SHIMMER motes. This battery has a minimum capacity of 250mAh. The voltage being supplied to the SHIMMER was a constant 3.701 volts for each test so the power consumption in Watts can be easily calculated if required.

III. Application Description and Results

In this section we will give a brief description of each application tested and discuss the major findings from each of the tests.

A. *Idle Test*

Null is a skeleton application that is distributed with TinyOS. It is the most basic TinyOS application possible which does nothing, with no interrupts or resources activated. As a result TinyOS puts the micro-controller unit (MCU) into low power sleep mode. This is useful to test the minimum power consumption of the Sensor node.

While performing these tests a bug was found in the TinyOS SHIMMER platform which resulted in very high power consumption by this application, approximately 7.5 mA when idle. This was due to the Bluetooth radio defaulting to being active. A bug fix was submitted to the platform maintainers and will be included in the next TinyOS release.

Table 2 shows the average current drawn by a SHIMMER mote running the Null application, with the above mentioned bug fixed, when no microSD card is inserted and when each of our six test cards are inserted. As can be seen in the table, by simply having the microSD card plugged into the mote the current being drawn in idle mode can increase from between 1.4 times to 12 times.

Compared with other sensor motes the power consumption of the SHIMMER, even when no microSD card is present, is relatively high at over 171 micro-amps. This can be explained by the presence of the Bluetooth radio. On current SHIMMERs the Bluetooth radio cannot be turned off but can only be set to what, for it, is a low power state. By physically removing this radio from the SHIMMER, i.e. desoldering it, the constant current draw drops by over two thirds when no microSD card is present. We include the power consumption of the Null application when the Bluetooth radio has been removed in table 2 in order to allow for comparisons with similar motes that only have an 802.15.4 radio. However, for the remainder of our tests we use a standard SHIMMER mote, i.e. one that has the Bluetooth radio attached.

TABLE II. Average Current Drawn when SHIMMER is Idle

	SHIMMER BT removed	**Standard SHIMMER**	
	Null	***Null***	***Null with SD init***
No microSD	56.743 μA	171.634 μA	386.896 μA
SanDisk 2GB	132.870 μA	250.975 μA	431.009 μA
A-Data 2GB	128.182 μA	247.820 μA	427.634 μA
SanDisk 1GB	852.922 μA	1.125 mA	1.220 mA
A-Data 1GB	361.037 μA	479.505 μA	615.156 μA
SanDisk 64MB	143.116 μA	261.007 μA	425.189 μA
Kingmax 64MB	1.953 mA	2.069 mA	615.913 mA

For the next test the Null application was modified slightly, the only change being to wire in the SD drivers. This causes the SD initialization command to be automatically called at boot up, which simply sends a command to the microSD card telling it to enter idle mode. Other changes include modifying the state of some of the MCU's pins and enabling a number of interrupts. As can be seen in the table this simple change has the effect of increasing the average current significantly, ranging from an 8 per cent to 125 per cent increase, except for the Kingmax 64MB card, which decreased by 70 per cent. This is the case because all of the microSD cards except the Kingmax card defaulted to their lowest power idle state when powered up but the Kingmax card had to be explicitly set to this state.

During this test the two A-Data cards occasionally got stuck in high power states, 815.206 micro-amps for the 1 GB card, 32 per cent higher than the normal state, and 21.601 milli-amps for the 2 GB card, a massive 50.5 times higher than normal. These anomalies were due to the A-Data cards being slower to power-up from a cold boot than the other cards, and was remedied by adding an extra short delay in TinyOS SD driver initialization sequence. Again this fix has also been submitted to the maintainers of the TinyOS SHIMMER platform.

Finally, for this test at least, the power consumption of the SanDisk 1 GB card was much more erratic than the other cards. For approximately the first half hour after powering up the current being drawn varied between 838 micro-amps and 1.141 milli-amps, eventually settling down to the value shown in table 2. This value is also much higher than any of the other cards, at least after the Kingmax 64 MB card has been successfully set to the idle state. The reason for this anomaly is unknown, but is possibly due the card itself being faulty. However, we encountered no problem when writing to, or reading from the card, in any device, so we continued to use this card in the remainder of our tests.

TABLE III. CURRENT DRAWN AND TIMING

	512B		1KB		10KB		100KB	
	1 min average current	*Time idle to idle*	*1 min average current*	*Time idle to idle*	*1 min average current*	*Time idle to idle*	*1 min average current*	*Time idle to idle*
Transmit 802.15.4	264.554 µA	3.285 s	372.304 µA	3.299 s	801.956 µA	4.076 s	10.485 mA	35.058 s
SanDisk 2GB	458.577 µA	2.534 s	469.869 µA	3.174 s	764.281 µA	3.381 s	2.085 mA	14.972 s
A-Data 2GB	473.590 µA	2.528 s	516.531 µA	2.894 s	982.375 µA	3.836 s	4.78 mA	15.193 s
SanDisk 1GB	1.186 mA	2.806 s	1.196 mA	3.220 s	1.377 mA	2.874 s	2.976 mA	14.927 s
A-Data 1GB	630.530 µA	3.204 s	646.210 µA	3.337 s	889.130 µA	3.225 s	2.071 mA	17.005 s
SanDisk 64MB	537.694 µA	3.439 s	510.660 µA	3.218 s	1.147 mA	2.954 s	4.120 mA	14.802 s
Kingmax 64MB	647.926 µA	3.179 s	651.846 µA	3.160 s	766.593 µA	3.012 s	1.78 mA	14.948 s

The conclusion that can be drawn from these two tests is that the simple act of plugging a microSD card into a SHIMMER can have a very significant effect on power consumption, even if no data is ever written to or read from the card. There also appears to be no consistency in the power consumption by the microSD cards. From our limited sample there does not appear to be a pattern among cards of the same manufacturer or among cards of the same size. Whether or not there is consistency between cards from the same manufacturer and the same size could not be determined, simply because we did not have two such cards to test.

B. Once Off Data Transfers

We next tested the power consumption over a one minute period of transferring a fixed amount of data, 512 bytes, 1 kilobyte, 10 kilobytes and 100 kilobytes, over the 802.15.4 radio and to the microSD cards.

For the radio test active message (AM) packets [12] containing 32 bytes of data were used and the packets were all sent as quickly as possible once the radio was turn on. 32 byte data packets were chosen as it is close to the default TinyOS maximum data size of 28 bytes while still allowing us to send data of sizes that are multiples of 512 bytes, the fixed SD block size in TinyOS, without requiring partially filled packets.

The data was written to the microSD cards in 512 byte blocks, also as quickly as possible.

Table 3 shows the average current drawn over the one minute period for each data size, as well as the time between the MCU waking up to send/save the data and returning to its idle state.

The first, and possibly the most important point to note from these results is that saving various amounts of data to the microSD cards does not scale linearly, i.e. when saving twice the data the power consumption does not double, even when taking into account the once off overhead of powering up and down the microSD card at the start and end of the cycle. These results indicate that writing a large amount of data rarely is much more power efficient than writing small amounts regularly, more so than expected.

The act of transmitting data over the radio, again discounting the overhead of powering the radio up and down, is also not linear. Efficiency improves going from 512 bytes to 1 kilobyte to 10 kilobytes but then gets disproportionally worse with 100 kilobytes of data. This is expected as the number of collisions increases with more data sent, especially as our tests were conducted in an environment that had a number of devices operating in the 802.15.4 radio's 2.4 gigahertz band.

Strangely some of the microSD cards can save 1 kilobyte of data, or even 10 kilobytes of data, faster than it can save 512 bytes. This is probably the case as microSD cards were designed and optimized for accessing much greater amounts of data than we are employing, being rated to transfer data at up to 25 megabits per second.

Another interesting point to note is that performance of a particular microSD card is not consistent across the different data sizes. For example the 2GB A-Data card comes in second best when saving 512 bytes of data, only being 15 micro-amps worse than the best, but is the worst at saving 100 kilobytes, significantly so. Similarly, the Kingmax 64 MB card is the worst performer with the smallest amount of data, but is by far the best at the largest amount. We assume this is the case because the internal operating details of the various microSD cards have different internal buffering schemes and have been optimized for different operations.

These results bring up an interesting point. When using other types of non-removable flash storage on embedded systems a detailed data sheet, with timing, power and other salient details, is available to the device's designer to allow the most appropriate part, with suitable characteristics, to be chosen for use. However, despite the very large number of different microSD cards on the market we were only able to find two such datasheets for microSD cards [13][14], neither of which cover any of the microSD cards we were able to acquire. As a result choosing a microSD card suitable for a particular application, i.e. the particular data rate, either requires testing all the available cards or else is down to chance.

C. 24 Hour Test

Our final tests were to determine the power consumption of realistic application scenarios.

Our test application sampled 4 sensors at 1 hertz, the 12 bit analogue to digital converter (ADC) producing a two byte

TABLE IV. 24HR AVERAGE CURRENT

Transmitting 802.15.4	Saving to SD	Saving to SD (big buffer)
228.917 µA	469.188 µA	445.582 µA

value per sensor. The next step was to either:

- Send all the sensed data over the radio to a listening base station. Once a packet was full it was sent. With 32 bytes of data per packet this resulted in one packet every four seconds.
- Save the sensed data to the microSD card. A 32 byte header was attached to each saved block for indexing, and querying purposes. Once a 512 byte buffer was filled it was written to the microSD card, resulting in write once a minute. After 24 hours all the data saved to the microSD card was read back and the minimum, maximum and average values for each sensor was calculated and sent to the base station over the 802.15.4 radio. We used the SanDisk 2 GB microSD card for this test as it was one of the better performers over all in our previous tests.

Table 4 gives the results of these tests. Assuming perfect conditions, a 250 milliampere-hour battery would last forty five and a half days when transmitting the data over the radio but would only last twenty two point two days when saving the same data to the microSD card.

Due to the relative simplicity of our applications we had a large amount of unused RAM on the SHIMMER mote. In order to test the potential saving of fewer but larger writes to the microSD card we made use of this extra RAM and wrote to the card in eight kilobyte chunks. This resulted in extending the battery life of the application by approximately one point two days.

This test showed that while sending data over the radio is often considered to have the highest power cost in wireless sensor networks, saving the data to local removable flash memory is not necessarily better, at least not for a single hop network. However if a mote equipped with such storage was used to locally aggregate and then process the data from a portion of a network that is many hops away from the base station, the power savings quickly become clear, possibly not for the node doing the aggregating, but for the network as a whole.

IV. CONCLUSIONS

All microSD cards do not perform identically, or even similarly under the same conditions. From our tests we can conclude that individual cards can work more or less efficiently depending on the parameters and that performance of an application using one SD card cannot be used to judge the performance of the same application using any other card.

When dealing with low data rates, less than 10 kilobytes a minute, it is more power efficient for a sensor mote itself, but possibly not the sensor network as a whole, to transmit the same data over a low power radio. However, when dealing with higher data rates the reduction in power consumption achieved by saving the data locally can grow very quickly.

ACKNOWLEDGMENT

The authors wish to thank the following for their financial support:

- SFI Research Frontiers Programme grant number 05/RFP/CMS0071
- The Embark Initiative and Intel, who fund this research through the Irish Research Council for Science, Engineering and Technology (IRCSET) postgraduate research scholarship scheme.

REFERENCES

[1] SD Association (2009), http://www.sdcard.org, accessed 30 March 2009.

[2] Realtime Technologies Ltd (2008), http://www.shimmer-research.com, accessed: 30 March 2009.

[3] Sun Microsystems Ltd (2008), http://spot-eflash.dev.java.net, accessed: 30 March 2009.

[4] Freie Universität Berlin (2008), http://cst.mi.fu-berlin.de/projects/ScatterWeb/hardware/msb/index.html: 30 March 2009.

[5] P. Sikka, P. Corke, and L. Overs, "Wireless sensor devices for animal tracking and control," in *29th Annual IEEE International Conference on Local Computer Networks*, Tampa, Florida, USA, 2004, pp. 446-454.

[6] *802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs).* (2003) New York: IEEE Standards Association.

[7] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister, "System Architecture Directions for Networked Sensors," in 9th International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 93-104, Cambridge, Massachusetts, USA, 2000.

[8] Sourceforge Inc. (2009), http://sourceforge.net/projects/tinyos/, accessed 7 May 2009.

[9] Agilent Technologies (2009), http://www.home.agilent.com/agilent/product.jspx?pn=66321D, accessed 13 July 2009.

[10] Agilent Technologies (2009), http://www.home.agilent.com/agilent/product.jspx?nid=-536902309.389074.00&cc=US&lc=eng, accessed 13 July 2009.

[11] Ultralife Corporation (2006), http://www.ultralifebatteries.com/documents/techsheets/UBI-5113_UBC581730.pdf, accessed 13 July 2009.

[12] P. Buonadonna, J. Hill, and D. Culler, "Active Message Communication for Tiny Networked Sensors", In *20th Annual Joint Conference of the IEEE Computer and Communications Societies*, Anchorage, Alaska, USA, April 2001.

[13] Kingmax Technology Inc. (2005), http://www.digitalspirit.org/file/index.php/obj-download/docs/sd/Kingmax_microsd.pdf, accessed 13 July 2009.

[14] Power Quotient International Co. Ltd (2006), http://www.pqi.com.tw/upload/download/micro%20SD(2).pdf, accessed 13 July 2009.