

# Understanding Energy Consumption of Sensor Enabled Applications on Mobile Phones

Igor Crk, *Member, IEEE*, Fahd Albinali, *Member, IEEE*,  
Chris Gniady, *Member, IEEE*, John Hartman, *Member, IEEE*

**Abstract**—Recent research in ubiquitous and mobile computing uses mobile phones and wearable accelerometers to monitor individuals’ physical activities for personalized and proactive health care. The goal of this project is to measure and reduce the energy demand placed on mobile phones that monitor individuals’ physical activities for extended periods of time with limited access to battery recharging and mobile phone reception. Many issues must be addressed before mobile phones become a viable platform for remote health monitoring, including: security, reliability, privacy, and, most importantly, energy. Mobile phones are battery-operated, making energy a critical resource that must be carefully managed to ensure the longest running time before the battery is depleted. In a sense, all other issues are secondary, since the mobile phone will simply not function without energy. In this project, we therefore focus on understanding the energy consumption of a mobile phone that runs physical activity monitoring applications and consider ways to reduce its energy consumption.

## I. INTRODUCTION

Recent trends in medicine indicate that automatic detection of physical activity (PA) is important and will enable new types of health monitoring tools that help people maintain healthier life-styles. PA detection has the potential to transform traditional medicine that relies on general assessment of clinical symptoms and signs to a more personalized approach that is tailored to the individual, based on their behavioral and physical history [1], [2]. Coupled with their ability to communicate with a remote monitoring center, the onboard processing and storage capabilities makes mobile phones an attractive platform for remote health care monitoring. Furthermore, the availability of low cost sensors makes health monitoring applications more viable and cost effective.

Currently, CPU processing and communication are the largest consumers of energy in sensors and mobile devices. We anticipate that typical ubiquitous health monitoring scenarios will involve both battery powered wearable sensors and mobile phones that process the continuous sensor data at relatively high frequencies. These scenarios highlight the need for energy management of both the communication and computational aspects of such a system.

We present a framework for energy monitoring that includes both the hardware and software setup, along with current observations from energy profiling of a mobile phone and a PA monitoring application. Our early findings can provide guidance for other researchers undertaking projects

in energy efficiency of portable applications. We discuss the challenges encountered during the construction of the setup as well as the current limitations of the presented framework. Furthermore, we present an energy profile of a mobile phone and a PA monitoring applications that can inform the design of future applications. Finally, we discuss future research that will aim to improve the energy efficiency of the hardware and match the application’s algorithms to the underlying hardware to achieve overall energy efficiency on the mobile phone platform.

## II. MIT WOCKETS

Existing commercial products that monitor behavior and physical activity [3] [4] [5] rely on proprietary hardware that is in some cases unwieldy and in others bears a prohibitive price tag. The shortcomings of the commercial devices have caused the house\_n lab at MIT to initiate the development of wockets [6], [7]: a new open-source platform for fine-grained and continuous monitoring of physical activities using wireless accelerometers and mobile phones. The goal of the wockets project is to make both the hardware and software robust, so that the system can detect activity continuously and can be deployed at population-scale on a variety of smart-phones and pocket PCs. The hardware platform consists of wearable, miniature, unobtrusive and low-cost hardware sensors that measure motion of different parts of a human body using accelerometers. The sensors send data to a mobile phone that detects the type, duration and intensity of physical activity. Currently, Bluetooth is used to send wockets data from the sensors to the phone as it is the only widely-used wireless protocol available on mobile phones. It is anticipated that Ultra Low Power (ULP) Bluetooth will become increasingly popular in the near future.

It is important to emphasize that the use of mobile phones for collecting and processing data at high data rates poses new challenges to energy management on mobile phones. Phones have limited battery life and therefore limited processing. Running a wockets application that continuously receives data at 180Hz will deplete the battery in a matter of hours. For example, HTC’s Touch Diamond with a fully charged battery runs the wockets software for 4-5 hours before the battery is depleted. The evaluation presented here was performed on a similar HTC TyTN II mobile phone with a comparable battery lifetime running the data collection software for the wockets. For practical deployments, the battery has to last for at least 12-16 hours, allowing the phone to be charged overnight.

I. Crk, C. Gniady, and J. Hartman are with the Computer Science Department, University of Arizona in Tucson, AZ 85719 USA. Corresponding author e-mail icrk@cs.arizona.edu

F. Albinali is with the House\_n Lab at MIT, Cambridge, MA 02142 USA.

a) *PA System Overview.*: The PA recognition system consists of wireless wearable accelerometer sensors sampling at 180Hz that run a Nordic proprietary wireless protocol, an intermediate device that receives data from the accelerometers and relays it via Bluetooth, and a WM6 smart-phone with Bluetooth that collects data from the intermediate device. It is important to note that the Nordic protocol is a low-power wireless protocol that allows accelerometers to operate at 180 Hz using a light-weight 3v cell battery for over 24 hours. Compared to Bluetooth chips, the Nordic chip is relatively inexpensive and employs a simple wireless protocol with neither flow control nor error recovery. The wockets system consists of multiple sensors that capture limb movements from multiple body points simultaneously. All accelerometers are 3-axial  $\pm 2g$  that provide sufficient sensitivity to capture the different intensities of body movements. The wockets software that initializes the Bluetooth connection and starts receiving data and calculates feature vectors every 500ms in real-time.

b) *Activity Recognition Algorithm.*: The algorithm that runs on the phone is a pruned C4.5 decision tree. The acceleration data streams are broken into 500ms sliding windows with 50% overlap. Data from each stream is independently interpolated using cubic splines to make data windows of equal size irrespective of the sampling rate and the number of accelerometers that are configured in the setup. A feature vector is then calculated that includes time and frequency domain features that capture variability, relative orientation of sensors, entropy, correlation to capture simultaneous motion of limbs, FFT peaks, and energy to differentiate intensities of activities.

#### A. Energy Management

Energy management is vital on portable and mobile devices since their functionality depends on a battery that has limited capacity. As a result, designers are looking to provide more energy efficient hardware and software. Hardware techniques are focusing on optimizing energy efficiency of displays [8], processors [9], communication hardware [10], software transformation and scaling [11], and energy efficient data aggregation and communication [12].

### III. PROFILING ENVIRONMENT

We evaluate the power consumption of the mobile environment with a combination of software and hardware tools: National Instruments 6009 Data Acquisition module (NI6009DAQ), power supply, IBM T42 laptop with LabView 8.6, and the HTC TyTN II Windows Mobile phone.

Figure 1 shows the profiling environment. The HTC TyTN II phone is connected directly to the power supply to provide constant voltage and reduce complexity of energy measurements. Since this phone, like most modern phones, require communication with the battery in order to operate, we have reused some of the electronic components from the battery pack and wired them directly to the power supply. The current draw is calculated by capturing the voltage drop across a  $100m\Omega$  current sense resistor wired into the phone's

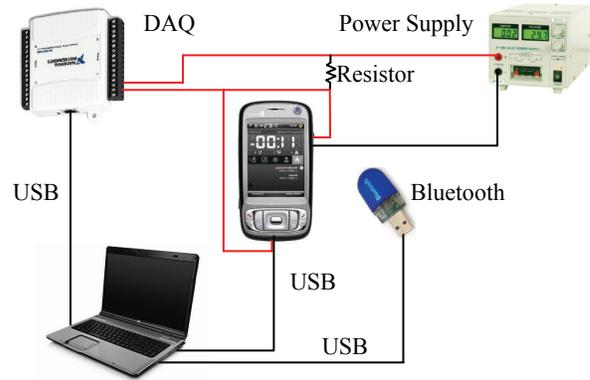


Fig. 1. Profiling environment

connection to the power supply. The power supply itself is set to 4V. This configuration allows the phone to operate continuously on the constant voltage and the current can be monitored for arbitrary lengths of time.

To allow synchronization and communication between the measurement VI and the phone we rely on the USB connection. However, the USB connection is also used to charge the phone and supplies energy rendering the initial setup inaccurate. To address the issue we cut the power supply wire in the USB cable and wire it directly to the power supply. The phone expects the positive wire to be energized before it establishes a USB connection and once it is energized, the phone is normally charged through it. To account for all energy consumption by the phone we measure current drawn by the power terminals inside battery as well as the USB connection as shown in Figure 1.

We selected LabView, a platform and development environment for the G graphical programming language, to obtain detailed measurements with external triggering. LabView monitors the NI6009 DAQ, which provides the voltage across the sense resistor. The analog input channel is sampled at 1kHz. In order to profile the physical activity recognition software running on the phone, we also instrument the application to trigger the start and stop of data acquisition in LabView via TCP. The communication between LabView and the phone is established via a USB connection, following the ActiveSync synchronization period.

Providing start and stop triggers allow detailed energy profiling of the application phases. We created Virtual Instruments (VIs) in order to manage incoming triggers. A trigger manager VI establishes TCP communication with the application and notifies the collection mechanism about the currently executing code segment. The data collection VI, records data and labels it with the given segment name. The subsequent end trigger stops data collection and the readings are stored in separate files named by the received code segment name and a counter value that separates the readings of the same code segment. When each sampling is stored in a separate file, the number of readings contained in the file combined with the sampling rate gives the runtime of the sampled code segment.

Component	Power (W)
Phone	.11
Screen (dimmed)	.43
Bluetooth (idle)	.01
Bluetooth (active)	.37
USB Sync (idle)	.54

TABLE I  
HTC TyTN II POWER CONSUMPTION BASELINE.

### A. Challenges

The challenges in this preliminary profiling setup revolve around the accuracy of triggering and ultimately the accuracy of the collected energy consumption of the profiled application. For example, the delays of TCP communication used for triggering the data collection result in excess samples being collected at both the beginning and end of the intended sampling period. However, the excess is not greater than half of the round-trip time for the trigger communication and synchronization packets, and in our case less than 1 millisecond, so less than the time to collect one sample using the 1kS/s sampling rate. In future work, we will increase the sampling rate and collect more accurate readings through other means of triggering the data collection.

Another challenge is the separation of power consumption by component. At this point, the setup allows simply the sampling of the power consumption of the entire unit, meaning that the different aspects of the profile are obtained by subtracting the measured consumption of the hardware and operating system from the power consumed while the profiled application is executing. Similarly, profiling the power consumption of the various hardware components of the unit, such as the screen, the Bluetooth transmitter, or the CPU and memory, are obtained by first measuring the power consumed while those components are idling or off. We will investigate more sophisticated black box measuring techniques in future work.

## IV. PHONE PROFILE

As described in section III-A, the phone profile is first obtained in order to establish a baseline. The basic power consumption of the phone is obtained by placing it flight mode, meaning no communication modules are active, including the main telephone transmitter, Bluetooth, and WiFi. Any running applications are removed from the task list and the screen is shut off. Once this part of the profile is established, the next step is to selectively switch the various phone components on, with the aim of establishing the power consumption profile of all the individual components and the environment in which the profiled application will be running. These measurements are detailed in Table I.

As is evident from the Table I, the phone itself, operating but idle, consumes less energy than any one of its components except in the case of an active but idle Bluetooth module. The TyTN II battery is assumed to hold 1350mAh of charge. The screen alone will consume almost 80% of the phone’s energy if it is the only active component. In other words, with a display that’s been powered on, but left in the state with the dimmest backlight, the phone consumes 5

times as much energy as it does when remaining flight mode standby.

An active but idle Bluetooth communication module has an almost negligible impact on the phone’s standby lifetime, reducing it by about a tenth of the normal battery life in this state. However, the application that was profiled relies on a constant stream of data arriving from a set of Bluetooth-connected accelerometers. Interestingly, optimization of the Bluetooth communications from the accelerometers is limited by the observation that, on the TyTN II, the Bluetooth module remains in an active state for 10 seconds following any Bluetooth activity. Additional work is required to verify that removal of this feature will not adversely affect Bluetooth communications while providing additional opportunities for energy reduction.

The power consumed while the USB connection is active is also profiled, since data collection requires this link to enable the triggering mechanism. Considering the nature of the profile physical activity application, a deployed system would not require this type of communication, so it is profiled simply to be discounted during application profiling and to inform the reader.

## V. APPLICATION PROFILE

Once the baseline profile is established, the running application can be similarly profiled. We have isolated several sections of code that are either executed frequently or involve an intensive computation. Table II shows the profiled code segments, detailing the average execution time in seconds, average energy consumed per execution, average power, and average frequency at which each segment was observed. BT INIT and BT PAIR represent a one-time initialization of the Bluetooth connection and pairing the phone with another Bluetooth device respectively. These segments consume more than .2 watts in periods exceeding 5 seconds. The overall cost of initializing the Bluetooth connection is minimal because it is invoked only once. The PA application’s requirement for constant incoming data also eliminates the option of shutting down Bluetooth communications.

EXTRACT computes machine learning features from incoming data. It includes calling FFT that calculates the Fast Fourier Transform. EXTRACT is called by GEN FV, or Generate Feature Vector, when enough wockets data accumulates. These functions form the bulk of the processing of the data arriving from the accelerometers. GRAPH and STORE are code segments respectively responsible for displaying and storing data. If we consider the feature extraction functions combined with GRAPH and STORE, assuming that the phone’s screen is off, we can expect to get about 10.5 hours of use before the battery runs out, when the screen is on the expected battery life is 5.7 hours. In this case, we are assuming that the phone and WiFi modules are not active and that the battery capacity is 1350mAh. Depending on whether the display is on or off, the profiled segments alone affect the battery life by as much as 7%.

Figures 2, 2B, and 2C show the breakdown of power consumption by major components. In Figure 2A we see that

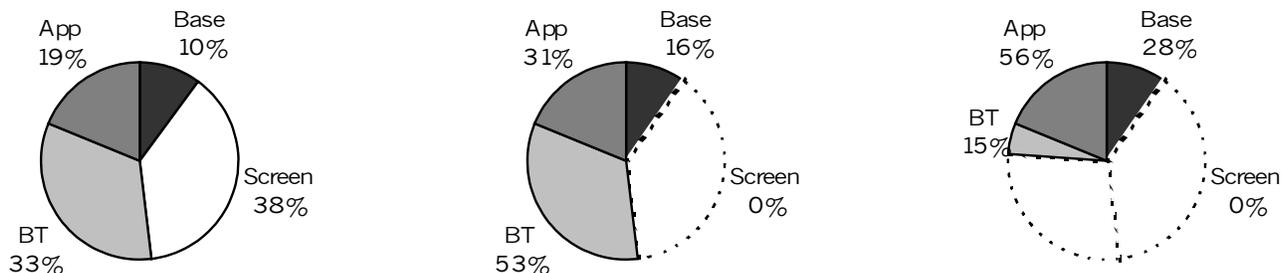


Fig. 2. Power consumption broken down by major components. Left (A) chart shows the power consumption breakdown when all components are active. Center (B) shows the relative power consumption when the screen is shut off. Right (C) chart shows power consumption when the screen is off and Bluetooth power consumption is optimized.

Segment ID	Time in Sec.	Ave. J	Ave. W	Times Seen per sec.
BT INIT	5.75	1.674	0.291	N/A (one time)
BT PAIR	4.68	1.214	0.259	N/A (one time)
EXTRACT	0.158	0.037	0.037	0.24
GEN FV	0.101	0.023	0.023	0.26
FFT	0.021	0.005	0.005	1.46
GRAPH	0.011	0.002	0.002	5.55
STORE	0.011	0.002	0.002	5.55

TABLE II

SELECTED CODE SEGMENTS OF THE PA MONITORING APP. PROFILE.

when the display is powered on and operating at the dimmest setting, it consumes 38% of energy when the PA application is running and Bluetooth is active, as required by PA. The phone alone consumes 10%, labeled as base and being the one aspect of the profile that is not considered for optimization in this work. When this environment is active, meaning screen is on and the application and Bluetooth transmissions are unoptimized, we can expect that the 1350mAh battery will provide 4.8 hours of operation.

Figure 2B shows the relative power consumption by major components when the screen is powered off. We see that in this case, Bluetooth operation consumes 53% of power, considering the remaining components. The simple action of powering the screen off for the duration of the application's activity increases the operation time to 7.7 hours. Next, we consider optimizing the Bluetooth operation.

The observed Bluetooth transmission behavior on the TyTN II is that the phone's power consumption remains high for some time following the transmission's completion. Given this overhead, we consider buffering the incoming sensor data on the PA system's intermediate device. The Bluetooth 2.0+EDS theoretical transmission rate is 3Mbits/sec. The intermediate device collects 180 samples each second, each of which is 5 bytes. Considering this data rate, and allowing for the intermediate device to buffer at least 360kB of data, the Bluetooth transmissions can be cut down to as little as a few seconds at a time separated by about 40 seconds of idleness, allowing the Bluetooth module to enter a low power mode. Figure 2C shows the breakdown of the remaining power consumption categories. We see that in this scenario, Bluetooth power consumption has been reduced to only 15% of the total. Under this load, the operation time of the phone increases to 9 hours. Application optimizations can further increase the operational time.

## VI. CONCLUSION

The constant delivery of samples collected from the sensors and passed on by the intermediate device keeps the bluetooth transmitting constantly and the PA application frequently computing feature vectors and managing the incoming data. Two avenues for optimization are immediately apparent for future research: optimizing the Bluetooth protocol or the applications use of Bluetooth, and optimizing the live data analysis. Currently, the Bluetooth communication and the application amount for over 50% of the energy consumed in the phone when the display is powered on, and since the display remaining on is not a requirement of the application this proportion is considerably greater when the display is off.

## REFERENCES

- [1] E. M. Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman, "Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor," in *Proceedings of The International Symposium on Wearable Computers*, 2007.
- [2] F. Albinali, N. Davies, and A. Friday, "Structural learning of activities from sparse datasets," in *Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications*, 2007.
- [3] Philips, "Actiheart," Online Specification, 2008, <http://actiheart.respirionics.com/>.
- [4] C. Health, "Oxycon mobile," Online Specification, 2008, [http://www.viasyshealthcare.com/prod\\_serv/prodDetail.aspx?config=ps\\_prodDtl&prodID=3](http://www.viasyshealthcare.com/prod_serv/prodDetail.aspx?config=ps_prodDtl&prodID=3).
- [5] ActiGraph, "Actigraph," Online Specification, 2008, <http://www.theactigraph.com/>.
- [6] MIT, "Open source accelerometers for phones," Online Specification, 2008, <http://web.mit.edu/wockets/>.
- [7] —, "Wockets," Online Specification, 2008, <http://wockets.wikispaces.com/>.
- [8] S. Iyer, L. Luo, R. Mayo, and P. Ranganathan, "Energy-adaptive display system designs for future mobile environments," in *MobiSys '03: Proceedings of the 1st international conference on Mobile systems, applications and services*. New York, NY, USA: ACM, 2003, pp. 245–258.
- [9] P. Pillai and K. G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," in *SOSP '01: Proceedings of the eighteenth ACM symposium on Operating systems principles*. New York, NY, USA: ACM, 2001, pp. 89–102.
- [10] M. Honkanen, A. Lappetelainen, and K. Kivekas, "Low end extension for bluetooth," in *Proceedings of the Radio and Wireless Conference*. IEEE, 2004.
- [11] R. N. Mayo and P. Ranganathan, "Energy consumption in mobile devices: Why future systems need requirements-aware energy scale-down," in *Lecture Notes in Computer Science*, 2003.
- [12] M. A. Sharaf, J. Beaver, R. Labrinidis, and P. K. Chrysanthis, "Tina: A scheme for temporal coherency-aware in-network aggregation," in *In Proc. of MobiDE*. ACM Press, 2003, pp. 69–76.