Demo Abstract: AudioDAQ: Turning the Mobile Phone's Headset Port into a Universal Data Acquisition Interface

Andrew Robinson
EECS Department
University of Michigan
Ann Arbor, MI 48109
androbin@umich.edu

Sonal Verma
EECS Department
University of Michigan
Ann Arbor, MI 48109
sonalv@umich.edu

Prabal Dutta
EECS Department
University of Michigan
Ann Arbor, MI 48109
prabal@umich.edu

ABSTRACT

Smartphone peripherals like the Square card reader, RedEye mini, and HiJack platform suggest a growing interest in using the headset port for more than just headsets. However, these peripherals only support sporadic activities in an efficient manner. Continuous sensing applications – like monitoring EKG signals – is possible but remains too inefficient for many realistic usage scenarios. We present AudioDAQ, a new sensor data acquisition platform. In contrast with prior work, AudioDAQ requires no hardware or software modifications on the phone, uses significantly less power, and allows continuous data capture over extended periods of time. The design is efficient because we draw all necessary power from the microphone bias voltage, and it is general because this voltage is present in every headset port. Data is modulated within the audible range, captured with the built-in voice recording app, and sent to a server for processing and storage. We show the viability of this approach by demonstrating an EKG monitor that can capture data continuously for hours.

Categories and Subject Descriptors

B.4.2 [**HARDWARE**]: Input/Output and Data Communications—*Input/Output Devices*; C.3 [**COMPUTER-COMMUNICATION NETWORKS**]: Special-Purpose and Application-Based Systems

General Terms

Design, Experimentation, Measurement, Performance

Keywords

Mobile phones, Energy harvesting, Phone peripherals, Audio communications, Participatory sensing

1. INTRODUCTION

Use of the mobile phone headset port as a peripheral interface is growing. A new class of devices including the HiJack platform [1] and RedEye mini [2] has recently emerged to exploit the ubiquitous nature of this port. However, many of these devices are only efficient for sporadic activities. Their power interfaces often harvest energy from waveforms continuously exported over the audio output channel(s) for power. Generating these waveforms often requires significant CPU time and prevents the phone from sleeping. As a result, the system as a whole consumes orders of magnitude greater power than delivered to the hardware peripheral itself, making the designs impractical for continuous or long-running applications.

IPSN'12, April 16–20, 2012, Beijing, China.
ACM 978-1-4503-1227-1/12/04.

Peripheral Device	Run Power	Sample Power	Sample Time
AudioDAQ	66.7 mW	107 mW	32 hr
HiJack	205 mW	304 mW	11.5 hr
RedEye mini	451 mW	451 mW	7.5 hr

Table 1: Power draw breakdown of RedEye mini, HiJack, and AudioDAQ. For sensors requiring small amounts of power AudioDAQ allows for extended sampling periods.

We recognize that there exists a class of sensors that require a continuous, yet minuscule, amount of power to operate over extended intervals. This class includes biometric sensors such as EKG monitors and environmental sensors, such as electrochemical gas detectors and soil moisture probes. These sensors are well-matched for use in mobile phones, which provide rich auxiliary data in the form of location services, orientation detection, and context via annotation from the end-user.

In this demonstration, we present AudioDAQ, a new platform for continuous data acquisition using the headset port of a mobile phone. Our system differs from current peripheral interfaces by drawing all necessary power from the microphone bias voltage, and by making use of the built-in audio recording application, encoding all data in an analog signal within the audio passband. These differences are key in achieving much lower system power consumption, as shown in Table 1, and making our design more universal among smart and feature phones. The microphone bias voltage typically used to power amplifying circuitry found in electret microphones is provided for "free" by the phone, requiring no CPU time to generate. The built-in audio recording app makes use of optimized hardware audio encoding facilities. Both features were found in every phone we surveyed, suggesting that our design maybe universal.

The recorded data could include overlaid audio annotations, and the combined data maybe sent via e-mail or MMS to a cloud-based processor where the original signal is reconstructed and optional audio annotations are extracted and added to the data.

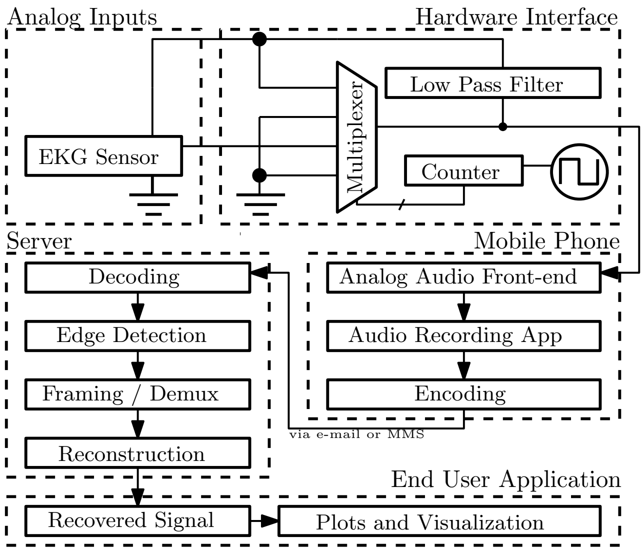


Figure 1: System architecture. By using the built-in audio recording utility, and offloading the processing to cloud-based servers the load placed on the mobile phone is significantly diminished and sample time is extended.

2. SYSTEM ARCHITECTURE

AudioDAQ is a low-power, continuous data acquisition platform for the mobile phone. It uses the headset port's microphone line to *both* power external sensors and acquire their data. Filters separate the power and data pathways from each other. A multiplexer circuit breaks up DC and low-frequency AC sensor signals to enable them to pass through the band-limited audio channel (with a typical passband of 20 Hz to 15-20 kHz). The multiplexed (or modulated) signal can be compressed efficiently (using a phone's voice memo application and native codecs) for subsequent processing offsite by remote servers. As seen in Figure 1, the architecture offloads the majority of the post processing to a remote server to allow for efficient data capture.

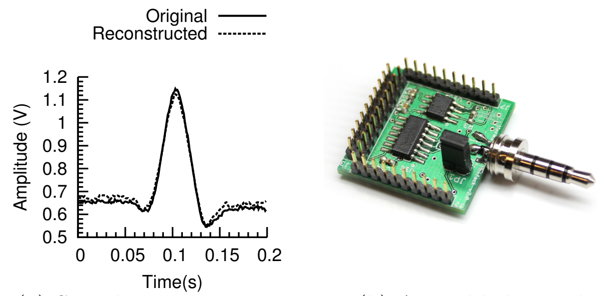
2.1 Acquiring Analog Sensor Data

The typical audio front-end can directly acquire signals in the 20 Hz to 20 kHz range with amplitudes between 1-25 mV. But, many signals have principal frequency components below 20 Hz, or consist entirely of a DC component, making them impossible to pass.

We use an analog multiplexer as a poor man's modulator. By switching at 1.5 kHz, a frequency within the passband of the audio system this allows the "modulated" signal to pass. Our system achieves a sample rate of 375 Hz.

We calibrate the signal by time-division multiplexing the microphone line between ground, the signal, and a reference voltage. By expressing the magnitude of the signal as a value between ground and the reference voltage, we can recover the original DC value. The simultaneous capture of multiple analog signals is possible via addition of channels to the multiplexer. Figure 2(a) shows the reconstructed signal overlaid with the original signal, scaled using the known reference voltage. This shows we can recover both AC and DC components of the signal.

By adding a momentary push-button disconnect switch between AudioDAQ and the phone we can temporarily enable the built-in microphone, allowing capture of voice annotations. AudioDAQ is distinctive enough from voice data that separation is easily done server-side.



(a) Signal Reconstruction (b) Assembled Board

Figure 2: (a) shows the overlay of the original signal, and the final output after feeding the signal through (b) AudioDAQ and reconstructing it on the server.

2.2 Cloud Processing of Voice and Sensor Data

Cloud processing of the data frees us from the limited processing capabilities of a mobile platform, and makes it possible to use the built-in voice recording application. This gives us greater flexibility in the signal processing algorithms we can employ and allows AudioDAQ to work universally on phones where custom software is impossible for budgetary or technical reasons.

Data can be sent via e-mail or MMS message to a server, where the original signal is recovered, voice annotations are extracted, and graphs are automatically generated and sent back to the user via e-mail for analysis. The data may be stored on the server for later analysis and use.

3. DEMONSTRATION

In this demo, we show the full operation of the AudioDAQ system. We capture a short 5-10s sample of EKG data from a user using the voice recording app on an iPhone attached to our system, shown in Figure 2(b), add a voice annotation at the end of the capture with the name of the participant, and then send this recording to our server for processing.

The data is reconstructed, voice data are extracted and processed using a voice recognition library, and finally a graph is constructed, containing the EKG waveform and the annotation data.

4. ACKNOWLEDGMENTS

This material is supported in part by NSF Awards #0964120 ("CNS-NeTS") and #1059372 ("CI-ADDO-NEW").

5. REFERENCES

[1] Y.-S. Kuo, S. Verma, T. Schmid, and P. Dutta. Hijacking power and bandwidth from the mobile phone's audio interface. In *DEV'10: Proceedings of the First Annual Symposium on Computing for Development*, Dec. 2010.

[2] ThinkFlood, Inc. RedEye Mini universal remote for iPhone, iPod Touch & iPad. `http://thinkflood.com/products/redeye-mini/`, Apr. 2011.