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

Sonal Verma, Andrew Robinson, and Prabal Dutta
Electrical Engineering and Computer Science Department
University of Michigan
Ann Arbor, MI 48109
{sonalv, androbin, prabal}@umich.edu

Abstract

We present AudioDAQ, a new platform for continuous data acquisition using the headset port of a mobile phone. AudioDAQ differs from existing phone peripheral interfaces by drawing all necessary power from the microphone bias voltage, encoding all data as analog audio, and leveraging the phone's built-in voice memo application (or a custom application) for continuous data collection. These properties make the AudioDAQ design more universal, so it works across a broad range of phones including sophisticated smart phones and simpler feature phones, enables simple analog peripherals without requiring a microcontroller, requires no hardware or software modifications on the phone itself, uses significantly less power than prior approaches, and allows continuous data capture over an extended period of time. The AudioDAQ design is efficient because it draws all necessary power from the microphone bias voltage, and it is general because this voltage and a voice memo application are present on most mobile phones in use today. We show the viability of our architecture by evaluating an end-to-end system that can capture EKG signals continuously for hours and send the data to the cloud for storage, processing, and visualization.

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

SenSys'12, November 6–9, 2012, Toronto, ON, Canada.

1 Introduction

Mobile phones have become ubiquitous in modern life and they provide many of the features that made personal computers popular, but in a compact form-factor, untethered to power, and always networked. Considerable research over the past decade has transformed the mobile phone into a platform that supports continuous sensing applications. Although many sensors – like accelerometers, gyroscopes, and imagers – have been integrated into the phone, many other sensors – like EKG, air quality, and soil moisture – have not. The desire to support such sensors, coupled with a limited set of direct-connect interfaces suitable for powering external peripherals and transferring data to and from them, have led some to search for a universal peripheral interface port. One candidate for such a peripheral interface is the mobile phone's headset port. This interface is mostly standardized, at least physically and somewhat electrically, across many mobile phones, by necessity to ensure compatibility with a broad range of hands-free and headphone audio devices. And, recently introduced peripherals like the Square card reader [24] and RedEye mini [29] suggest a growing interest in using the headset port for more than just headsets.

For reasons of cost, simplicity, and ubiquity, using the headset port as a general-purpose interface for transferring power and data to peripheral devices is an attractive option. However, there is a considerable variance in the pass-band characteristics, power delivery ability, signal pin outs, and microphone bias voltage among headset ports on mobile phones. This leads us to conclude that contrary to recent claims [12], the headset port is not as universal as one might hope. For example, designs built to work with the iPhone may fail to work on many Android or Windows phones, and vice versa. Furthermore, designs built to work with smartphones may fail to work with the far more numerous but less capable feature phones, making current approaches brittle.

The headset port imposes several limitations on mobile phone peripherals. The power transfer through the headset port is inefficient. Its pass-band characteristics severely limit the frequency range of passable signals. Features that at first seem to be of great utility, such as the high sampling rate, and the ability to create arbitrary waveforms on the output audio channels, turn out to be poorly suited for simple peripherals that do more than essentially capture or playback audio.

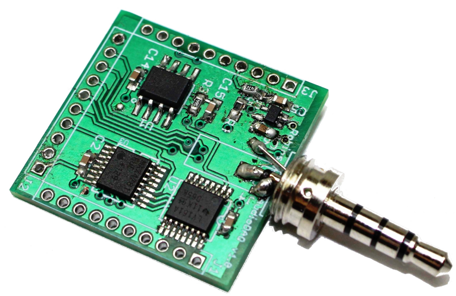


Figure 1: AudioDAQ, a square-inch mobile phone peripheral interface capable of providing a power and communication channel to a variety of sensors by (a) modulating an low-rate analog signal into the audio pass-band and (b) harvesting a trickle of power from the microphone bias voltage to power external sensors and circuits. AudioDAQ is inexpensive ($5) in modest volumes (10k), makes use of common electronic components (oscillator, counter, multiplexer, and regulator), and can be miniaturized easily through integration.

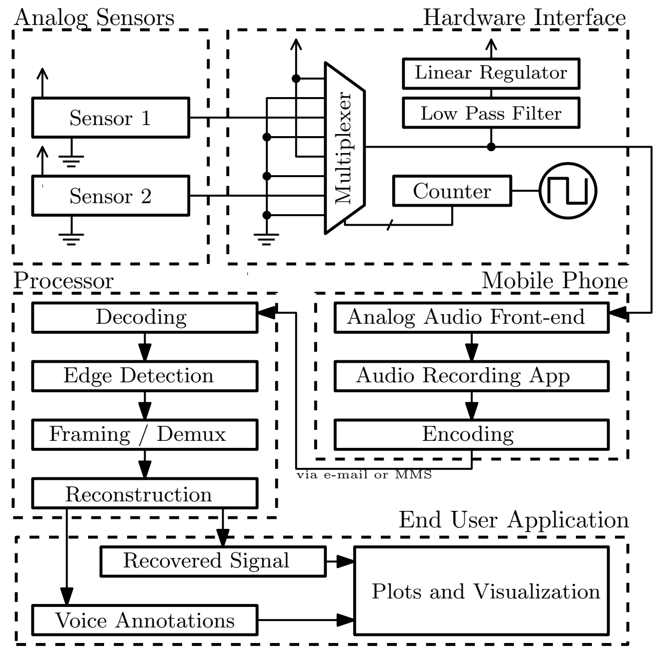


Figure 2: System architecture. By using the built-in audio recording utility and offloading the processing to cloud based servers, the load placed on the mobile phone is significantly reduced and the sample time is extended.

Recent work on creating peripherals for the headset port has focused on maximizing power delivery for sporadic sensing activities [12]. However, we claim that what works for sporadic activities does not scale well to continuous sensing. There exists a class of sensors that requires a continuous, but minuscule, amount of power to operate over extended periods of time. Some examples include bio-metric sensors like EKG monitors and environmental sensors like electrochemical gas detectors and soil moisture probes. By targeting this subclass of sensing peripherals, we limit the power delivery and data transfer requirements, giving us greater flexibility when designing around the headset port.

In this paper, we present AudioDAQ, shown in Figure 1. AudioDAQ is a system that offers continuous capture of analog sensor signals and provides power to the sensors that is drawn from the microphone bias voltage. We present an example application of a low-power, two-lead EKG sensor capable of acquiring cardiac data continuously for extended sampling periods. Next, we evaluate our design, characterize signal recovery and distortion across diverse operating conditions, and compare the energy costs of signal processing on the phone vs processing on a remote server. Next, we discuss the limitations and future directions of AudioDAQ. Finally, we justify our design with a brief survey of existing headset port energy harvesters and discuss the trade-offs that each design point offers.

2 Design

AudioDAQ is an end-to-end system for capturing, annotating, storing, sending, and processing analog sensor data. Figure 2 presents the overall system architecture. The system consists of four major blocks:

Hardware Interface. It includes a linear regulator to ensure that external sensors and circuits are supplied by a stable voltage source. The analog output signal from a sensor is modulated into the audio pass-band using a simple modulation scheme.

Mobile Phone. The modulated signal outputted by the hardware interface is fed into the next block, the mobile phone, which acts as a encoder and a storage device. Inside the phone, the signal is conditioned by the analog audio front-end and is captured using the built-in voice recording application allowing for extended capture periods. The signal is compressed using the built-in audio compression utilities for efficient storage. The data are next transmitted to a remote server for processing.

Processor. The encoded audio data are received by a remote server and decoded. On the server the data goes through multiple stages of processing: normalization, edge-detection, framing, and reconstruction. If more than one signal has been multiplexed into the composite waveform, it is demultiplexed at this point. In some cases, it is also possible to eliminate the encoding step and implement the processing functions directly on the mobile phone.

End User Applications. Finally, the sensor data reaches the data application layer of the architecture. Here any implementation-specific transformations are performed and the data are formatted for use by the end user. In a typical application, this stage will process the data, extract domain specific key metrics, and generate useful plots. In our example application of the EKG sensor, this step extracts the heart-rate and plots the EKG.

2.1 Microphone Bias as Power Source

Recent work in creating peripheral devices for the headset port has focused on harvesting energy from the audio output driver. Since the audio output driver is designed to drive earphones and speakers, it can deliver many milliwatts [12].

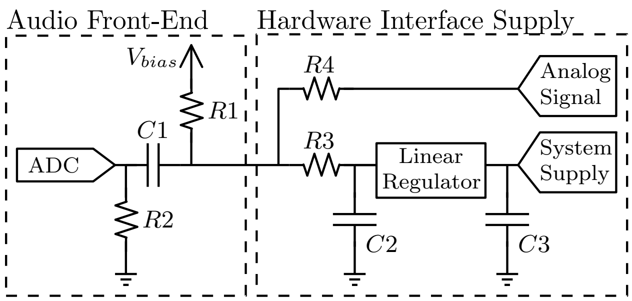


Figure 3: Microphone harvester schematic, including a model of the phone audio front-end which shows both how the microphone bias voltage is generated, and the filtering done on the input signal.

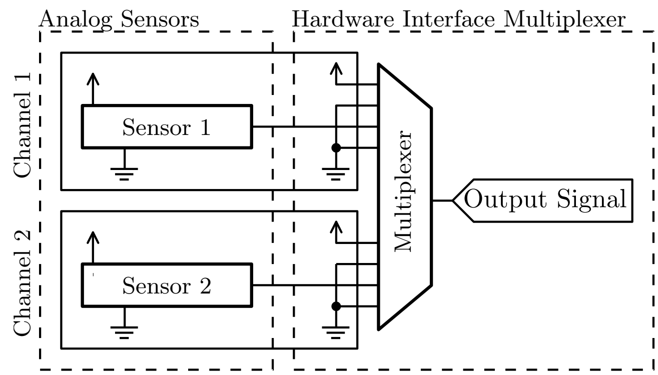


Figure 5: Acquiring analog sensor data using 8:1 analog multiplexer. We switch between analog inputs at a frequency of 1.2 kHz to create a composite signal within the audio passband. To recover the DC value of the analog signal a reference voltage of a known value is added to the analog multiplexer inputs. Due to the high-impedance nature of both the reference voltage and the analog sensor signal, every other input is connected to a low-impedance ground source, which reduces bias charge from building up on the mobile phone's internal high-pass filtering capacitor.

However, using the headset port as a power source for peripheral devices presents significant design challenges. While the audio output driver is capable of sourcing tens of milliamps, it does so at a voltage lower than typically required to operate power electronics. Further, it requires this power to be transferred as an AC signal in the audible frequency range. This approach was used in HiJack [12]. Software on the phone generates a sinusoidal waveform that is sent to the audio output driver and exported over an audio channel. Next, the signal is fed through a micro-transformer to reach a voltage level high enough to switch a FET. Finally, the signal is rectified and regulated to a stable DC voltage. However, using the audio output driver has significant drawbacks. Custom written phone software is required to generate the sinusoidal waveform which draws substantial power on the phone. Moreover, converting the output of the audio driver into a usable DC voltage requires inefficient rectification circuitry. Finally, while the typical audio driver can deliver a significant amount of power compared with the microphone bias voltage, there is a high degree of variability between phones, making it difficult to design a circuit that is universal enough to work across many headsets.

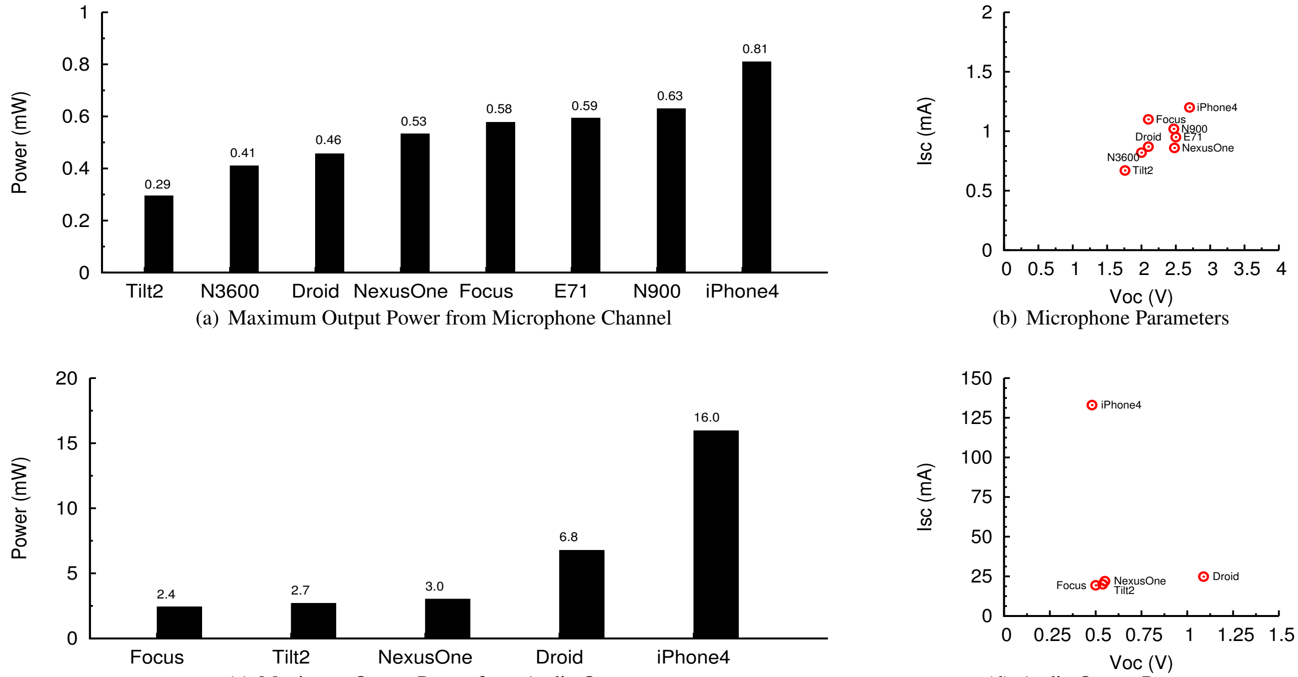
In this study, we explore the limits of using the microphone bias voltage to power AudioDAQ, and any attached sensors and circuits. We are aware of only one other contemporaneous system that uses the bias voltage to power active electronics in this manner, used to interface an Android phone with an amateur radio [10]. The microphone bias voltage is intended to power only the small amplifying FET found in electret condenser microphones and is only capable of delivering small amount of current. AudioDAQ consumes approximately 110 μW, well below the maximum power delivery of a typical headset port.

Figure 4 shows the maximum deliverable power and the optimal point in the P-I-V space for both the microphone bias voltage and the audio output driver. Fewer phones were surveyed when measuring the audio output driver's parameters due to the difficulties in developing custom software for each phone. The open-circuit voltage of the microphone bias line in the phones surveyed ranges from 1.7 V to 2.7 V. AudioDAQ requires 1.8 V to operate, making it compatible with nearly all of the phones we surveyed without requiring voltage boosting circuitry.

Using the microphone bias voltage as a power source offers a new set of design challenges. Since the microphone channel is used both to power sensor and also to transmit sensor data back to the phone, the power supply and data transfer characteristics are deeply coupled. Figure 3 shows a model of the phone circuitry responsible for processing data from the sensor and generating the microphone bias voltage. $R3$ and $C2$ form a single order RC filter to stabilize the linear regulator and prevent regulator control loop noise from reducing the fidelity of the analog signal from the sensor. This signal is of extremely low amplitude (10mV peak-to-peak) to make it compatible with the phone's audio processing circuitry. The cut off frequency for this low-pass filter is set to 50 Hz which is far below the modulation frequency of the analog signals. The microphone bias voltage is a relatively high-impedance voltage source and its output current is limited by R1, as shown in Figure 3. Therefore, components cannot draw even modest transient currents without proper bypass capacitors. Otherwise, large voltage drops will result. However, the capacitance must be kept small enough to ensure that they do not bypass the modulated signal itself.

2.2 Acquiring Analog Sensor Data

The typical audio front end in a mobile phone is optimized to acquire signals with amplitudes around 10 mV peak-to-peak, and audio frequencies in the 20 Hz to 20 kHz range. However, many signals either have principal frequency components below 20 Hz (e.g. EKG signals) or are purely DC in nature. This makes it difficult or impossible to pass them through the band-limited channel. To overcome this limitation, we use an analog multiplexer as a simple modulator to encode the analog signal into the audio passband by rapidly switching between signal and ground, as shown in Figure 5. The analog multiplexer is driven from a counter and clocked with an RC oscillator at 1.2 kHz.



(a) Maximum Output Power from Microphone Channel

(b) Microphone Parameters

(c) Maximum Output Power from Audio Output

(d) Audio Output Parameters

Figure 4: Output characteristics of the audio interface on several mobile phones. The maximum output power of the microphone bias line, as calculated from the short circuit current and open circuit voltage in 4(a) is much lower on average than the maximum output power deliverable via the left and right audio channels 4(c). Fortunately it is high enough to power many small active transducers and the circuitry required to modulate their output signal into the audio pass-band. The significantly lower variability in maximum output power across phones, and the optimal current-voltage point (4(b) vs. 4(d)) required to achieve this power across many phones makes designing power electronics for this channel easier.

We expect the analog signal from the sensors to be a high-impedance voltage anywhere between system ground and the reference voltage of 1.8 V. The magnitude of this signal is too large to be fed directly into the microphone line. To fit within the 10 mV limit, we add a scaling resistor between the output of the multiplexer and the microphone bias line. This resistor is identified as R4 in Figure 3. Our calculations indicate that sizing the resistor around 200 kΩ scales the signal appropriately into an amplitude range that does not overwhelm the audio processing electronics for the mobile phones that we surveyed.

Variability in the microphone bias voltage causes variations in the amplitude of the multiplexed signal. The mapping of signal amplitude to ADC counts among headsets is inconsistent, with each phone having a slightly different scaling factor when capturing an audio signal. These variations make it impossible to directly recover the absolute magnitude of the analog signal. To estimate the absolute voltage of analog input signals, we add a reference voltage generated by the linear regulator to the multiplexer. This effectively time-division multiplexes the output signal from the multiplexer between ground, the analog signal, and a known reference voltage. This allows us to later recover the absolute ground-referenced DC value by scaling and shifting the analog signals with respect to the reference and ground voltages.

The connections to the input of the multiplexer shown in Figure 5 are, in order, a voltage reference, ground, analog signal, and ground (again). Switching to a low impedance system ground after each voltage signal helps remove residual charge buildup on the capacitor C1 of the high-pass filter on the mobile device.

The final step in creating a flexible analog input design is allowing for the simultaneous capture of multiple input channels. We realize this feature in our design by simply duplicating this four signal block on the multiplexer for each additional channel we wish to capture. Our present design, shown in Figure 5, enables simultaneous capture of two channels. If it is necessary to capture just a single channel, the two inputs are tied together with an on-board jumper.

2.3 Power Efficiency

AudioDAQ uses an efficient linear regulator. However, the input filters add some resistance to the power path which results in some amount of power loss. However, these filters are necessary for separating the signal and power components that share the microphone bias line. Since AudioDAQ draws only 110 μW, its efficiency is nearly inconsequential when compared with the other subsystems in the mobile phone. It is more important to consider the design decisions that influence the draw of other subsystems like the CPU and storage which have more impact on battery life.

To better understand how the design of AudioDAQ influences the total power draw of the mobile phone, and to avoid optimization of subsystems which have relatively minor contributions to the overall power budget of the system, we perform several experiments on the HiJack platform [12]. We used the publicly available source code and selectively disabled parts of the system to measure the approximate power draw of each subsystem. The results from these experiment are summarized in Table 1. While these numbers are specific to HiJack and the iPhone, we expect that they generalize to other platforms and devices.

From Table 1, we can see that, with the exception of the screen, which can be easily disabled by pressing the power button on most mobile phones, there is no clear candidate for optimization, so we sidestep the question of optimizing a particular subsystem.

By choosing to use the microphone bias voltage to power our system instead of the audio output driver, we eliminate power required to generate the output audio waveform and reduce the power required for I/O to the audio channels. Therefore, we allow the phone to keep a large portion of the audio acquisition interface inactive.

By encoding sensor data in the audio pass-band and simply recording it for later processing, we reduce the power required to process the input signal. This is possible because of the efficient codec hardware accelerators found in many mobile phones.

Subsystem	Power (Avg)
Display Screen	333.0 mW
Audio Channel IO	114.0 mW
Input Signal Processing	106.0 mW
Base System	68.5 mW
Output Signal Generation	60.0 mW
Application Overhead	59.5 mW
Data Visualization	19.1 mW

Table 1: Power draw of the HiJack platform on an Apple iPod, measured by selectively disabling certain components of the software system via direct modification of the source code. With the exception of the screen, no single component dominates the system power budget, making simple, single, or localized optimizations ineffective.

2.4 Capturing and Storing Data Efficiently

For long-term data acquisition, data capture and storage must be efficient. We do not usually process the data on the phone, so the entire audio data must be stored. Storing raw data would be space prohibitive, so we employ the built in compression utilities found on the phone. Almost all mobile phones come bundled with a voice memo application that makes use of these algorithms to record low-quality audio suitable for voice memos.

On iOS devices, the voice memo application stores data in Advanced Audio Coding (AAC) format, which is a standard widely used by Apple at 64 kbps. Samples are taken at 44.1 kHz from a single channel. On Google Android phones, the built in application uses the Adaptive Multi-rate (AMR) encoding with an 8 kHz sample rate by default. Many other formats including AAC are available as part of the API, and sound recording applications that produce higher quality records do exist. Many feature phones also use the AMR encoding because it is specially designed for mobile applications and has been widely adopted.

All these codecs can sufficiently compress audio into file sizes practical to store on a mobile phone. Both smartphones and feature phones often come with built-in hardware support for these compression algorithms. On iOS and Android devices, specific media subsystems are exposed to the developers that allow for hardware-enhanced encoding. On feature phones, the CPUs often have special multimedia modules. The implementation of the codecs is done either completely in hardware or in heavily optimized low-level programming languages. Therefore, storing the audio data is efficient across most phones, and codecs do a good job of compressing the raw audio data into reasonable file sizes.

2.5 Processing Sensor Data

The original signal is typically extracted from the multiplexed analog sensor data on a remote server. The audio files are uploaded to this remote server (but could be processed on the phone) via e-mail where they are immediately processed and the data are extracted. Most feature phones and smartphones manufactured today have the software capabilities to record and transfer a voice memo, making the AudioDAQ design quite universal for sensor data capture.

2.5.1 Signal Reconstruction

The signal reconstruction algorithm is implemented in Python, making use of external libraries to perform codec specific decoding of the compressed audio files. It is designed to examine data in a rolling window to allow for both online and offline processing. It is also robust to noisy input data because it relies only on local data points and simply discards signals that are too noisy for proper reconstruction.

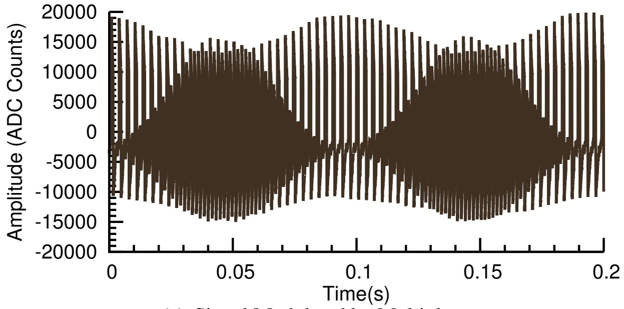
The simplicity of the hardware interface block in our design of AudioDAQ poses a challenge for the signal reconstruction algorithm. Since the analog multiplexer can send no channel delimiter, the framing information must be implicitly determined. Signal reconstruction in the AudioDAQ system occurs in five stages:

Decoding. The audio encoding format is deduced from the file extension and encoding specific magic bytes. The appropriate decoder is run on the data and the raw audio information is extracted.

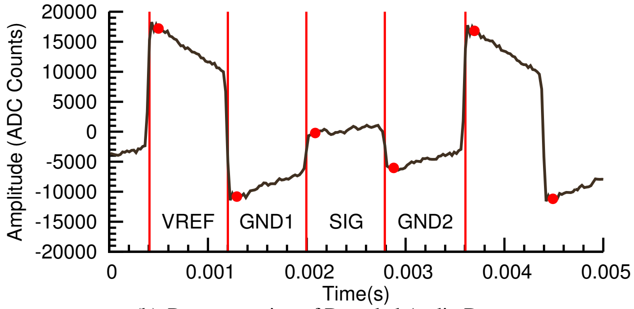
Edge Detection. The transition edges that are created when the multiplexer switches signals are detected and marked on the data set.

Value Estimation: The regions between the edges are evaluated. Extreme outliers are discarded and an estimate of the value of the signal in that region is obtained.

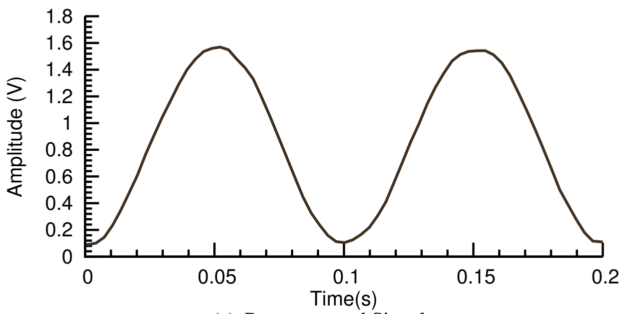
Framing. Based on previously processed frames, the framing is deduced and tracked. Framing is important to determine which values corresponds to ground, the voltage reference, and the actual analog signal. Each frame of input data consists of the four multiplexer signals as discussed in Section 2.2.



(a) Signal Modulated by Multiplexer



(b) Reconstruction of Decoded Audio Data



(c) Reconstructed Signal

Figure 6: The raw signal captured by the phone's audio front-end, recorded, transmitted to a computer, and decoded as a sine wave 6(a) is fed into the reconstruction algorithm. Edge detection (the vertical red bars) and value extraction (the red dots) takes place and is shown in 6(b), where a small segment of the signal has been magnified and annotated with the debug output of the algorithm. The extracted values are then used to compute the original input signal 6(c). A small, periodic drifting bias in the absolute value of the signal can be observed in 6(a). This bias comes from residual charge building up on the high-pass input filter's capacitor (Figure 3, C1) inside the mobile phone. The same effect can be seen on a smaller time scale in 6(b), where after switching the signal follows an exponential drift towards zero-offset.

Calculation. The absolute analog signal voltage for the frame is calculated by expressing the signal value as a point between the ground and voltage reference value and then shifting and scaling the voltage with respect to the known voltage reference and ground.

A secondary benefit of including the ground to reference voltage transition is that it gives a reliable and repeating signal to help frame our data. Since the analog sensor signal could be at a value close to ground, it is impossible to reliably detect the edge transition between the signal and ground. In a frame the only two transitions we can reliably detect are the transition from the previous frame's ground to the reference voltage, and the transition from the reference voltage to ground. These transitions are detected by finding the maxima and minima of the first-order numerical derivative of the signal. The distance between these two transitions is calculated and used to estimate the final two transitions of the frame between the unknown analog signal and ground. The vertical bars in Figure 6(b) show the detected and calculated edge transitions for a short period of input signal.

After edge detection, the regions between the edges are evaluated. The high-pass filter capacitor starts to immediately affect the signal after switching, so the left-most region of the signal is used to estimate the nominal value of the signal for each region. A small number of points are averaged to a single value. These values are also plotted in Figure 6(b).

Finally, the analog signal value is extracted. Up until now all processing has been done with ADC counts. To obtain the actual real-valued voltage, we express the analog signal as a value between the voltage reference and ground. The modulation scheme produces two ground values per frame. We use the average of these two values. After obtaining this value, we can then multiply it by the known reference voltage (1.8 V in our system) to obtain the original analog signal.

2.5.2 *Offline Processing*

The most common scenario for data processing involves collecting data to a compressed file and sending the file to a server for post-processing. This has the advantage of mitigating the power cost of transmitting the data to the remote location by delaying it until it is convenient for the operator, such as when near a charger or connected with a faster wireless network like 802.11b, or docked with a desktop.

An alternative offline processing scheme that was considered but not explored involved recording the data using the voice memo hardware and then periodically reading it in and doing a high-performance, faster than real-time computation on the data. This batch processing avoids the high idle cost of the CPU and the high wakeup cost of going from sleep to wake up mode and still offers near-real-time performance. Since a major strength of the AudioDAQ system is its compatibility with almost any hardware without requiring additional software, we chose not to explore this option.

2.5.3 *Online Processing*

A less common scenario involves processing the data in real time. This is useful for demonstrative purposes where the sensor data are wirelessly transmitted to a remote host for real time display. Assuming a sufficiently fast connection is available, it is possible to stream data to a remote host. Audio encoding algorithms for VoIP systems such as Speex, which has a mobile port available, make this possible over TCP. Even a simple telephone call could provide the bandwidth necessary to stream the data. Streaming in real time would dramatically reduce the battery life due to the greater power demands of the wireless radio in the mobile phone.

Pinout	Tip	Ring 1	Ring 2	Sleeve
(a) 3.5 mm	L	R	G	M & C
(b) 3.5 mm	L	R	G	M
(c) 2.5 mm	L	R	M	G
(d) 2.5 mm	S+	M+	S–	M–

Table 2: Audio headset pinout of several common mobile phones. (a) Apple, Blackberry, HP, Samsung; (b), (c) Nokia smartphone headset connector; (d) Nokia headset socket. Left audio output (L), right audio output (R), microphone input (M), control (C) and ground (G) are common on many smartphone headset ports. Some simple phone headsets offer speaker positive (S+), speaker negative (S–), microphone positive (M+), and microphone negative (M–). Although many smartphones use the same headset interface, there are important pinout differences between device classes, and even within some manufacturer's product lines, showing that the ubiquitous headset port is not universal.

2.6 Capturing Voice Annotations

An obvious addition to an analog sensor capture system is a method to annotate the samples. Since we are limited by the phone's built in voice recording application, we do not have the ability to allow for the user to input text directly to be stored alongside the collected data. However, we can collect voice annotations using the same application that we use to collect the sensor data by alternating data and voice in the captured audio files, in effect recursively time-division multiplexing the signal.

Mobile phones typically detect the presence of a microphone by the DC voltage present on the microphone bias line. Most microphones pull the voltage of the bias line down past a certain threshold. When the phone detects this, it automatically shifts between the internal microphone and the externally connected peripheral. We exploit this behavior by adding a momentary disconnect switch to the peripheral, effectively allowing the user to pause data collection to inject and record a short audio annotation in-band (and in-line) with the collected data.

Since AudioDAQ has a distinctive principle switching frequency, it is easy to algorithmically detect the difference between voice data and analog sensor data. This is done server-side. Next, using open-source voice recognition libraries, the speech is converted into text which is paired with the reconstructed data.

2.7 Mechanical Design

We chose a 3.5 mm headset jack because it has become the standard for smartphones with the introduction of applications such as music playback that make use of external headsets. Further, the mapping of pins to logical signals in 3.5 mm headset jack implementations is more consistent when compared to the now less common 2.5 mm interface for which we found two common, yet separate mappings. Table 2 shows the different pinout configurations for audio jacks across various phones. If required, inexpensive adapters and "proprietary" headset connectors are available to connect 3.5 mm peripherals to 2.5 mm ports [5].

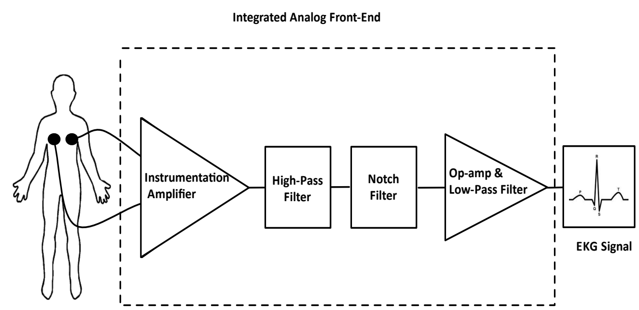


Figure 7: Block diagram of the EKG sensor system. Amplification is done in two stages to prevent the large amount of noise from pushing the signal too close to the supply rails, and destroying the EKG signal in the process. A micropower differential amplifier first amplifies the signal received via electrodes placed on two body locations and removes common mode noise. This amplified signal is then passed through a notch filter to remove 60 Hz interference ambient power line noise. The final operational amplifier provides an appropriate gain, amplifying the signal to a voltage level usable by AudioDAQ and biasing it to half the supply voltage.

We chose a square-inch form factor for AudioDAQ and a pinout that is mechanically compatible with existing HiJack sensors, although most of the HiJack I/O lines are left unconnected. If necessary, the circuitry could be made more compact or even built into a single integrated circuit and incorporated into a molded audio headset jack itself. The present square-inch form factor gives a good trade-off between small size and ease of development and debugging.

2.8 Low-Power EKG Sensor

Mobile devices have the potential to seamlessly blend health care into our daily lives [8, 18]. Evolving mobile health care technology can help professional care-givers to better monitor their patients, detect problems earlier, and reduce the need for hospital visits which are often expensive and inconvenient [19, 31]. In addition, mobile health care can empower individuals to better monitor and manage their own health, encouraging them to live healthier lifestyles and prevent many health problems before they begin by providing methods for early detection. For these reasons, we chose to develop a low-power, low-cost, portable EKG monitor which interfaces to mobile phone using AudioDAQ and illustrates the key operating principles. This battery-free sensor enables monitoring of an individual's cardiac activity for extended periods of time using relatively inexpensive electronics and existing mobile phones. It allows for data collection across a wide variety of phones, and for transmission of the data to a remote location where it is analyzed by automated algorithms for abnormalities, or by doctors for diagnosis.

The EKG sensor is a two lead device, with the leads attached to the subject's body using conductive hydrogel electrodes either on the left and right arm or wrist or directly across the chest. The signal is passed through two stages of amplification with active filters in between to remove noise as shown in Figure 7.

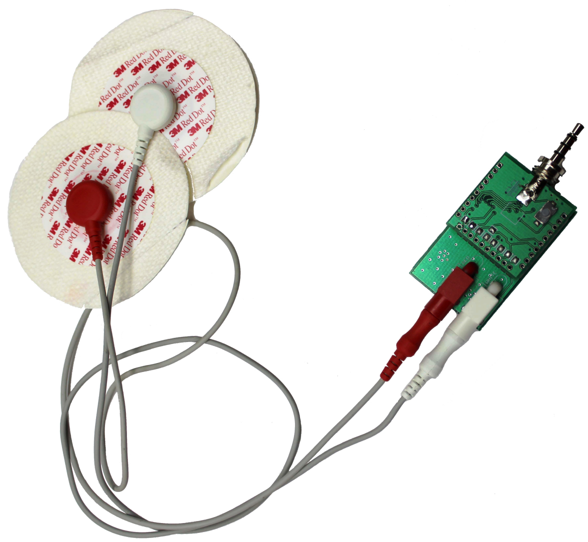


Figure 8: The AudioDAQ board interfaced with a 2-lead EKG sensor. The sensor draws 120 μA at 1.8 V and provides the necessary filtering and amplification.

Amplifying cardiac signals, which are in the range of -5.0 mV to +5.0 mV [30], and filtering out stray environmental noise captured by the human body poses a significant design challenge which is made more difficult by the power budget constraints imposed by the AudioDAQ system. Instrumentation amplifiers were chosen with exceptionally low-power draw. The first stage of amplification uses a differential operational amplifier which has a high common mode rejection ratio of 95 dB and rejects common mode noise found across the entire human body leaving only the differential signal from muscle contractions. It has a gain factor of five. It is integrated with a high-pass feedback filter that dynamically corrects DC shift in the ungrounded differential signal captured from the body. The human body acts as a large antenna around modern electrical grids. Therefore, the amplified signal is passed through a notch filter designed to remove 60 Hz common mode interference and noise. Finally the signal is fed through the last operational amplifier which also acts as a low-pass filter. It amplifies the filtered signal with a gain of twenty to bring it into the realm of voltage amplitudes commonly seen by analog to digital converters and to be used by AudioDAQ.

The useful bandwidth of an EKG signal is between 0.05 Hz and 150 Hz. We design the high pass and low-pass filter which act together as a band-pass filter configured to this frequency range. The output EKG signal is biased at approximately half of the supply voltage using a voltage divider. This minimizes the possibility that a large voltage spike (that can occur when the heart muscles contract) will be clipped by the operational amplifiers operating at their supply rails. The amplifier gains are configured such that with good skin-to-lead conductivity and lead placement, the EKG signal will have an amplitude of approximately 500 mV at the operational amplifier's output line.

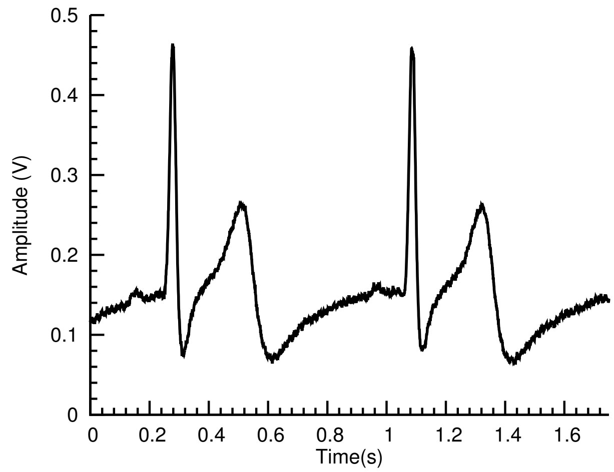


Figure 9: EKG trace of an individual. The 450 mV signal amplitude shows that all critical features (P wave, QRS complex, T wave) of a cardiac signal are preserved.

The EKG sensor module is interfaced with AudioDAQ to deliver the resulting EKG trace. Figure 9 shows the real-time EKG waveform of an individual captured with this system. A typical cardiac cycle (heartbeat) consists of a P wave, QRS complex, and T wave, all of which can be seen in our trace.

The EKG sensor interfaces with square-inch AudioDAQ base platform and draws only 216 μW. Table 3 shows the cost breakdown of an EKG sensor module. The total cost is $24, of which two-thirds is spent on leads which could be commoditized at larger volumes for a significant cost reduction.

Description	Mfg.	Part No.	Cost
Diff Op Amp	TI	INA333	$2.23
Op Amp	TI	OPA2333	$1.86
Wire Leads	WelchAllyn	008-0323-00	$16.0
Sockets	PlasticOne	8R003661901F	$3.00
PCB	Sunstone	n/a	$0.26
Passives	n/a	n/a	$0.75

Table 3: EKG sensor cost breakdown including the cost of the PCB board, major active components, and a small dollar value for passive components such as resistors and capacitors. The entire assembly costs approximately $24 in bulk, with the majority of the cost going towards specialized wire leads, which at larger production volumes could be commoditized and priced lower.

3 Evaluation

In this section, we evaluate the performance of the AudioDAQ system. We examine the major claims of universality and extended data collection lifetime. We also evaluate the system's power delivery capabilities, signal distortion characteristics, and cost effectiveness. Finally, we justify our design decision to process data remotely by comparing the power draw of local vs remote processing.

Phone	Works	Power Sourced	Power Drawn	Battery Capacity	Battery Life (est.)
iPhone4	Yes	322 μW	149 mW	5254 mWh	35 hr
Nexus One	Yes	168 μW	267 mW	5180 mWh	19 hr
Galaxy Nexus	Yes	247 μW	301 mW	6475 mWh	21 hr
N900	Yes	120 μW	270 mW	4884 mWh	18 hr
E71	Yes	90 μW	189 mW	5550 mWh	29 hr

Table 4: Results of testing the AudioDAQ system across a subset of mobile phones. The platform works on every phone tested, showing that it is universal across our test set, with variations in the maximum power delivery and total estimated lifetime.

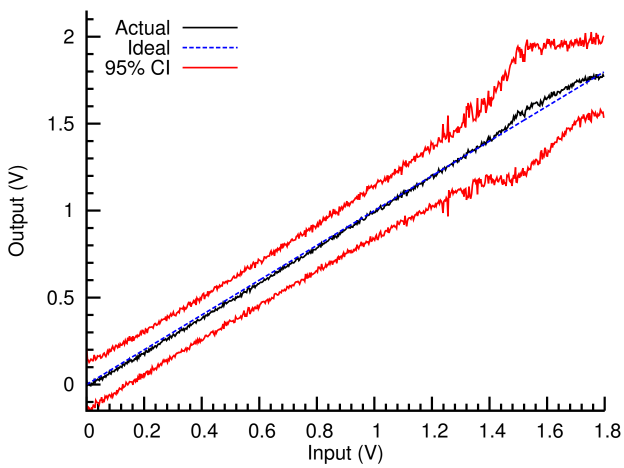


Figure 10: A voltage sweep was performed from ground to the reference voltage (1.8V) to determine the linearity of the reconstructed signal across the entire input voltage range. The results of the sweep were collected for 130 runs and the mean was found. A 95% confidence interval was computed. For applications where accuracy of the reconstructed signal is critical, keeping the range of the sensor between 0-1.2 V reduces the variance in sampled data.

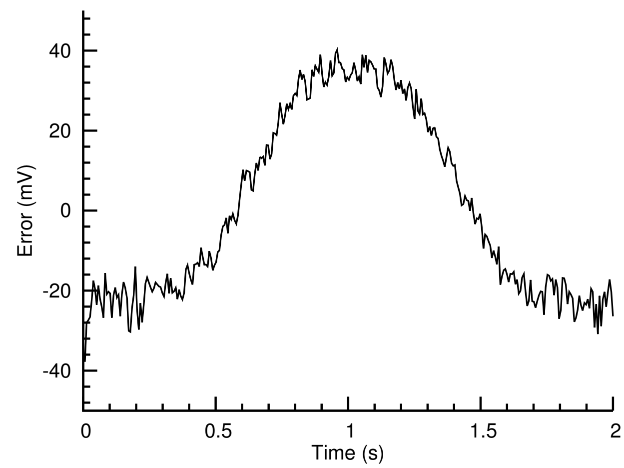


Figure 11: The error component of a constant-valued 0.9 V DC signal when multiplexed with a sinusoidal wave with a period of two seconds. The error was calculated from the mean value over 23 sinusoidal periods and is graphed over the period of the wave. The signals are weakly coupled to each other, with a maximum observed average error of 40 mV seen at the peak of the wave, representing a crosstalk-induced 4.4% error.

3.1 System Functionality

AudioDAQ works with a wide range of phones, provides a small amount of energy suitable to power a range of analog sensors, and allows for an extended battery life. In Table 4, the AudioDAQ system is evaluated against three smartphones and two feature phones. The system operates on all five phones, varying only in the lifetime and available power for sensor peripherals. Signal reconstruction was also performed on these phones and no significant difference was seen in accuracy, despite differences in audio recording circuitry and recording format.

Depending on the specific audio encoding used, the phone generates between 48 to 600 kB of sensor data per minute to be stored and eventually transmitted. For short data collections of up to 10 minutes in length, data files took a few minutes to transfer over a modern cellular network. All phones had sufficient storage to hold these files and mobile networks were sufficiently fast enough to quickly transfer them. Extended capture intervals required large amounts of storage, and were only practical to be transmitted over a Wi-Fi connection or via a physical USB cable. Many current phones have gigabytes of internal storage and Wi-Fi connectivity, making long-term captures viable. In return for a larger file size, we achieve compatibility with a large number of handsets which do not have alternatives to built-in voice data capture applications. While inconvenient, the large file size did not prevent data recovery. And, on smartphones, it is possible to perform data extraction from the raw audio signal, which dramatically reduces the data set size.

3.2 Collection Period

An extended capture was done on the iPhone4 to verify the estimated sample periods. The sample time for each phone is a function of both the efficiency of recording audio and the battery capacity. During the capture, at the 13 hr 30 min mark, the recording stopped. We restarted a second capture and allowed the phone to continue for another 13 hr 30 min period, at which point the capture ended again. We assume that this is a hard file size limit built into the audio recording application. The total capture lasted 27 hrs, with enough battery remaining to verify that the captures were successful and to transfer the files to a computer over a Wi-Fi connection. The audio files were 400 MB in size each.

3.3 Power Delivery to Load

AudioDAQ includes an ultra low-dropout linear regulator which regulates the microphone bias voltage to provide a stable power supply to external analog sensors. This regulator consumes only 2.5 μW while achieving an efficiency of 66% and an output voltage ripple of less than 5 mV in typical operating conditions, despite the high output impedance of the microphone bias voltage. The maximum deliverable power depends on the microphone bias characteristics of the host phone, with the voltage quickly dropping below the required 1.8 V as the power required by the load increases beyond a few hundred micro-watts. The AudioDAQ system draws 110 μW, with the remaining power headroom made available for use by analog sensors, which is shown in Table 4 across a variety of smartphones and feature phones.

The EKG sensor presented in Section 2.8 draws 216 μW. While the AudioDAQ system itself worked on all tested phones, the EKG sensor was only able to successfully work on two of the five phones in Table 4. This sensor was chosen as an example application because of its design complexity, cardiac data's sensitivity to noise, relatively high-power budget and requirement of high fidelity data delivery to the phone. With additional work, it may be possible to further reduce the power draw of the EKG sensor, making it compatible with a larger set of phones. As a second example application of AudioDAQ, we designed and prototyped a demonstration board that includes both an ambient light and temperature sensor. It has a power draw of only 58 μW and could be successfully powered by all tested phones.

AudioDAQ applications are not limited to just the two sensors that we implemented in this paper. Rather, they include a class of low-power devices including bio-metric sensors, environmental sensors, gas detectors, soil moisture probes, and many more.

3.4 Power Draw from Phone

Table 4 presents the system power draw across the tested phones. This power draw is the most significant factor affecting the total sample period. The iPhone4 had the lowest power draw, most likely due to the compiled nature of its applications and the design focus put into ensuring hardware accelerated multimedia processing. On both Google Android and the Nokia feature phones, applications are JIT compiled and run in a Dalvik or Java virtual machine, which results in lower efficiency. The iOS developer's API specifically makes hardware-assisted encoding methods available, and focuses on documenting how to effectively minimize power draw while processing audio. Google Android APIs are not as forthcoming, with hardware-assisted encoding being a platform dependent option, and no clear documentation on best practices for streamlining audio recording.

To justify our decision to process data server-side, we implemented the signal reconstruction algorithm on iOS and measured the power draw. While running signal reconstruction on the phone and storing the results, we found the phone used 203 mW, which is greater than the 149 mW used when simply recording the signal. However, these figures do not include the cost of transferring data to a server, which would favor processing on the phone and transferring a substantially smaller dataset.

3.5 Signal Recovery and Distortion

The majority of analog sensing applications involve reading a slowly-varying DC signal with great accuracy. AudioDAQ is well-suited for sampling a wide range of analog signals. To characterize the linearity and accuracy of the system's transfer function, we performed a slow voltage sweep with a period of 1 s from ground to the supply voltage. The results of this sweep are presented in Figure 10 and show that AudioDAQ is linear throughout the entire voltage range. The confidence interval was also calculated, and shown to be fairly constant until a voltage of 1.2 V. For applications that require high accuracy, we recommend limiting the sensor voltage to a range of 0-1.2 V to minimize the variance in sampled data. This can easily be done with a voltage divider.

To demonstrate AudioDAQ's ability to simultaneously capture multiple signals and to determine if any coupling between the channels exists, we performed the following experiment. We fed a sinusoidal wave with amplitude of 1.8 V and a period of 2 s through the system on one channel, while we feed a constant DC signal valued at half the regulated supply voltage through the second channel. The goal of the experiment was to evaluate the reconstructed value of the constant DC signal and check for any variance corresponding to the sinusoidal wave with time. Figure 11 shows the magnitude of the error introduced in the constant DC signal over the period of the sinusoidal wave and averaged across 23 periods. The error introduced into the DC signal never exceeded 40 mV, demonstrating a relatively modest degree (4.4%) of crosstalk between the two adjacent channels. Whether this materially affects the signal quality depends of course on the application.

When evaluating our system with the EKG demonstration board, the 300 Hz sampling rate was sufficient to capture all the important features of the EKG waveform and to provide a faithful signal reconstruction. Software was developed to extract the subject's heart rate from the raw data and plot it alongside the EKG waveform.

3.6 Hardware Cost Breakdown

Table 5 shows the cost breakdown for AudioDAQ. The hardware in AudioDAQ exclusively uses analog components that have been in mass production for years. These components are not high-performance devices and do not have any significant costs associated with them, making the basic AudioDAQ design rather inexpensive at just $5 for 10 k units. The cost includes the headset plug and printed circuit board.

Supporting a server to store and process the audio data from AudioDAQ devices can create overhead and costs, but the algorithm is reasonably efficient, running at approximately 20x real-time on a single core modern machine, allowing for a single server to support a large number of AudioDAQ devices sporadically sending data files.

Since AudioDAQ eliminates the need to write custom software for every platform, this results in significant savings in development, support and maintenance time and costs, largely offsetting or amortizing the costs incurred in maintaining a single server for processing.

Description	Mfg.	Part No.	Cost (10k)
Plug [11]	Kobiconn	171-7435-EX	$0.92
Regulator [28]	TI	TPS79718	$0.50
RC Timer [15]	Maxim	ICM7555	$0.13
Counter [27]	TI	SN74LV161	$0.10
Mux [16]	Maxim	MAX4781	$1.77
Switch [9]	C & K	KSR223	$0.50
PCB	Sunstone	n/a	$0.30
Passives	n/a	n/a	$0.75
Total			$4.97

Table 5: AudioDAQ hardware cost breakdown. The basic unit costs about $5 for 10 k units.

4 Discussion

In this section, we discuss the technical challenges that we faced in designing AudioDAQ to work with the mobile phone's headset port. The fundamental coupling between power and data when transferred simultaneously, but in opposite directions, over the microphone channel, and the constraints of the channel itself, significantly affect the design of AudioDAQ. As AudioDAQ was designed, a clear trade-off between signal reconstruction fidelity and maximum power delivery emerged.

4.1 Limitations

The AudioDAQ system is limited by the microphone channel used for data transmission and energy harvesting. The microphone line is not designed to support an arbitrary hardware peripheral. Rather, it has been optimized to achieve the low cost while efficiently providing the minimal power and bandwidth required to extract a voice signal from an external condenser microphone.

The significance of the trade-off between creating a better design by adding additional active components that could filter signals better, and keeping the system power draw low to provide more headroom for peripheral sensors became clear. The low power budget – just a few hundred microwatts for AudioDAQ and external sensors – forced us to adopt simpler passive techniques to solve filtering and signal conditioning issues at the small expense of signal fidelity.

4.1.1 Power Budget

The maximum power that can be harvested from the microphone channel varies from 0.29 mW to 0.81 mW, depending on the phone. This does limit the applications of AudioDAQ to the sensors which require low power to operate. Still the power is sufficient to support a wide variety of biometric and environmental sensors such as gas detectors and soil moisture probes etc., including the simple analog circuitry on the AudioDAQ board. Since there are many different sensing applications that require small, passive transducers and relatively simple amplifying circuitry, this does not severely limit the usefulness of the system within the chosen design space. However, some active sensors like air quality sensors and gas detectors which have substantially greater current drives for heaters or pulsed infrared lamps cannot be powered by AudioDAQ.

The energy that can be harvested is constrained by both the high impedance of the microphone bias voltage (set by an internal resistor R1 that can be seen in Figure 3) and by the design of the power supply. The power requirements of our system are below the optimized design space of many commercially available power supplies, making it difficult to find devices that can deliver reasonable efficiencies. This problem is exacerbated by the fact that we can tolerate very little ripple on the input supply rail. Most power supplies are designed provide a stable system output voltage, even if significant noise is present on the input rail. In our system, a quiet noise floor on the input side of the regulator is more important than having a extremely stable output voltage. The modulated signal from sensor which has extremely low amplitude is injected directly on to the microphone bias voltage. Any noise on the microphone line adds significant ripple to this signal and quickly makes it unrecoverable.

It has been shown from prior work that harvesting energy from a mobile phone is challenging to get right and presents a number of fundamental design problems. In early prototype versions of AudioDAQ, we tried modulating a signal across the audio channel using an external battery. Harvesting devices such as solar cells could have been a possible solution. However, they would add bulk to the AudioDAQ system without delivering significantly more energy making it worthwhile to harvest energy from the headset port.

Harvesting energy from the audio output driver is another viable solution which was also employed in HiJack. The power deliverable from the audio output driver is much greater than what can be obtained from the microphone bias voltage as shown in Figure 4. However, delivering power from the audio output driver requires custom software to be written for each feature phone or smartphone sacrificing the universality of the platform. In addition, this approach increases the power draw of phone significantly.

4.1.2 Coupling of Power and Data

There is a fundamental coupling between the amount of power that can be delivered and the magnitude of noise introduced in the data channel. The microphone bias voltage source for powering our system has relatively high impedance. The measured value for the biasing resistor that sits between the microphone bias voltage and the microphone channel is between 2 kΩ to 2.5 kΩ. This makes the voltage heavily dependent on the system load. Transient spikes caused by the switching of digital components can create large ripples in the input voltage. This problem is further exacerbated by the small amplitude of the signal (10 mV peak-to-peak) required to successfully pass through the audio front-end of the phone.

The obvious solution is to add additional bypass capacitance to the circuit to smooth the noise introduced by switching circuits. Placement of these bypass capacitors is critical for the modulated signal. If the capacitors are not properly isolated from the microphone channel, they will effectively create a low-pass filter and attenuate the signal. To isolate the decoupling capacitors from the microphone channel, a low-pass filter as seen in Figure 3 was added at the cost of increasing resistance and sacrificing the total power output.

4.1.3 *Sampling Rate*

The microphone channel is generally designed to support sampling frequencies that can capture signals within the audible range of human hearing (20 Hz-20 kHz). The pass-band characteristics of the microphone channel limit the sampling frequency for acquiring the analog sensor data. The compression scheme used in many mobile phones further limits the range of captured frequencies. While the iPhone uses a lossless encoding scheme, many other phones do not. Some phones employ adaptive multi-rate encoding (AMR) to compress voice data, which has a sampling frequency of 8 kHz, and is highly optimized for capturing the vocal range of the audio spectrum (80 Hz-1100 Hz). The AudioDAQ system is optimized for this frequency range with the multiplexer switching at a frequency of 1.2 kHz, which sits at the far right end of the vocal frequency range.

The number of samples required to reconstruct the original signal is limited by the Nyquist rate. The lowest frequency compression schemes operate at a sampling rate of 8 kHz, giving us approximately 7 samples for each multiplexer signal which is the lower limit of the number of samples required by the reconstruction algorithm to properly recover the original signal. Operating at higher frequency would first stress the limits of the recovery algorithm and finally push against the Nyquist limit.

For high-end smartphones like the iPhone that are often capable of capturing audio at high sampling rates (e.g. 44.1 kHz), the switching frequency can be increased. However, doing so will proportionally increase the power drawn by the system and will make it unusable for phones with slower sampling rates.

To preserve the universality of AudioDAQ, we keep the power draw low. Further, to ensure that the data transfer is optimized for compression schemes that favor vocal audio, which are more likely to be used on common mobile devices, we keep the multiplexer clocking frequency at 1.2 kHz. Since recovering a single analog signal requires at least four values, as explained in Section 2.5, the effective sampling frequency is 300 Hz (switching frequency divided by a factor of four). Capturing multiple channels requires time-division multiplexing each signal which further reduces the sampling frequency.

4.2 Future Work

The biggest limitations of AudioDAQ are the relatively small amount of power that can be delivered, and the effect it has on the fidelity of the recovered signal. The present power supply design involves a passive first order filter which we do not claim to be optimal. Noise from the power supply regulation loop and from digital circuitry, account for the error in the recovered signal. A more effective filter could reduce this noise and result in even better analog signal recovery. This filter would help in both reducing the ripple in the microphone channel created by the power supply and digital switching circuitry, and prevent the signal generated by the multiplexer from in turn creating feedback loops and destabilizing the circuitry generating it. For example, an active filter could yield a better response, but would introduce additional complexity into the design, and require sacrificing a portion of the power budget itself.

Another area of future improvement involves the power supply. In order to drive commercial logic gates, we have to maintain the microphone bias voltage above 1.8 V, which is not optimal. The power delivery of the microphone bias voltage could be improved by increasing the load and allowing the voltage to drop lower. Boosting this voltage back up by introducing a switching regulator could increase the maximum power delivery capabilities of the system. Early designs of AudioDAQ utilized such a switching supply. However, the noise introduced into the microphone channel was unacceptable and severely affected the fidelity of the recovered analog signal. Commercially available boost regulators operate in discontinuous mode to conserve power which result in large transient current spikes when the regulator switches. Due to the high-impedance of the microphone bias voltage, these current spikes cause a large voltage spike. Designing an efficient switching supply without this noise problem could provide more power for sensing peripherals. Power supply design traditionally focuses on reducing output ripple. Keeping input ripple low, especially when designing for a high-impedance voltage source, is a desired design parameter, and few existing solutions target this goal.

Finally, combining all of this circuitry into a single piece of silicon could significantly lower the power draw of AudioDAQ. Creating a chip small enough to be placed directly in the headset port plug itself would miniaturize the peripheral size and further expand the capabilities of the system.

5 Related Work

The AudioDAQ architecture, design, and implementation has been influenced by a large body of prior commercial products and academic projects focused on developing mobile phone peripherals [6, 14, 18, 23]. This body of work includes audio headset peripherals [2], wired/wireless peripherals [3], and vital signs monitors [1]. AudioDAQ is motivated in part by the opportunity to leverage the billions of already existing mobile phones in the world today (many of which are in developing regions).

5.1 Audio Headset Peripherals

HiJack is an open source energy harvesting platform for iOS devices [12]. It harvests power from the audio output driver making it capable of providing 7.4 mW to a load, with an efficiency of 47%. It allows for bi-directional communication between the phone and an on board MSP430 microcontroller at 8.82 kbps using Manchester encoding. HiJack requires generating the sinusoidal waveform from custom written phone software which consumes significant amount of power. AudioDAQ's novelty lies in its ability to harvest power from the microphone line of the mobile phone and use the same microphone line to enable continuous low power sensing. Contrary to HiJack, it enables sensing applications not just on iOS devices but on smartphones and feature phones. Further, it does not require any hardware/software modifications on the phone. It improves on HiJack by extending the sampling period, a direct result of simplifying the design by trading the flexibility of a microcontroller for a more power-efficient analog solution after recognizing that such a system is adequate for a large class of sensing applications.

Similar to HiJack, RedEye Mini [29] also uses the audio output driver for driving an LED that acts as a TV remote control. It integrates an upgradeable microcontroller and similar power harvesting circuitry. It provides a significantly higher power to the load by utilizing both left and right audio output channels, but as a result drains the phone's battery faster. Table 6 compares the power draw and estimated sample period of HiJack, RedEye Mini, and AudioDAQ on an iPhone4. Although, both HiJack and RedEye Mini can provide significantly more output power, it is provided at a cost to overall system power draw. Such high output power is not necessarily needed for a large class of sensing applications and makes them impractical for long-term data collection on mobile phones.

Square [24] is a credit card reader which interfaces to Android and iOS devices through the 3.5 mm audio jack to allow for mobile payments. Based on our own tear down, we found it detects the voltage spikes (30 mV peak-to-peak) generated by oscillating magnetic field created when a credit card is pushed through the magnetic read head. It has a robust algorithm to extract the data and uses error correction schemes to improve detection rates. It is a completely passive device requiring no power from the headset port. We borrow from it the idea that a simple hardware design can be augmented with robust algorithms to achieve complex functionality.

Other audio headset peripherals include My TV Remote [17], a simple purpose-built television remote control lacking the complexity of RedEye Mini; SwitchScience SoftModem [25], a general purpose development board for implementing communication mode similar to HiJack; iData [4], a PIC-based platform with limited technical details available; and H4I [22], a development platform also using communication mode similar to HiJack offering integration with existing commercial devices. All these peripherals show that there is interest in the headset port. However, each of them differs significantly from AudioDAQ. My TV Remote is much simpler than AudioDAQ, only allowing limited control of an infrared transmitting diode directly connected to the audio output channel. The SoftModem, iData, and H4I all provide bi-directional communication but none are self-powered or optimized for extended data collection.

Huyett's AFSK interface for Android smartphones allows the phone to interface with an amateur radio transceiver using a circuit built entirely from discretes – resistors, capacitors, diodes, and transistors – and be powered from the bias voltage [10]. This contemporaneous work shares the use of the microphone bias voltage to power external electronics.

5.2 Non-Headset Peripherals

Several designs have also been created to enable external sensor peripherals to communicate with mobile phones that do not make use of headset port, instead opting to use Bluetooth, USB, or other interface options that are standardized but are either not always available, not open, or not able to support a slave device.

FoneAstra [8] is a sensing platform developed by researchers at University of Washington in collaboration with Microsoft Research. It is a low-cost, programmable hardware plug-in that enables sensing applications on low-tier mobile phones. The system consists of an ARM7 microprocessor that connects and communicates to phone using serial I/O. It can be powered either from mobile phone's battery or by using a separate external battery.

Peripheral Device	**Run Power**	**Sample Power**	**Sample Time**
AudioDAQ	109 mW	149 mW	35 hr
HiJack	247 mW	346 mW	15 hr
RedEye mini	493 mW	493 mW	10 hr

Table 6: Power draw breakdown of RedEye mini, HiJack, and AudioDAQ. For sensors requiring small amounts of power AudioDAQ allows for extended sampling periods.

Little Rock [21] proposes a new architecture for mobile phones where the sampling and processing of sensor data is offloaded to a secondary low-power microprocessor. This approach allows the primary processor to go into sleep mode, enabling long-term sensing. If successfully integrated into new mobile phones, Little Rock would achieve longer sample intervals with much greater efficiencies. However, AudioDAQ has the advantage of working with the phones of today, requiring no hardware modifications.

Low-power Bluetooth (BT) may emerge as a popular standard for connecting sensors wirelessly to mobile phones in the future. However, low-power BT may only be limited to smartphones and not support feature phones. More importantly, low power BT would still require an external energy source to power the sensor peripherals. AudioDAQ is different from these systems because it harvests energy from the microphone line of the phone to power the sensor peripherals, thereby avoiding an external power source.

5.3 EKG Monitors

There has been considerable research in making EKG monitors increasingly low-power, wearable and portable [7]. Our example application of EKG sensor builds on the prior efforts in this area.

L. Fay et al. built a micro-power EKG amplifier [13]. A remarkable feature of this amplifier is a significantly smaller power draw of 2.8 μW. However, the circuitry is implemented on a chip, making it prohibitively expensive if not mass produced. Our EKG sensor is built completely from widely available off-the-shelf components and is assembled on a standard PCB.

The wireless EKG system built by T. R. F. Fulford-Jones et al. can be attached to a Mica2 wireless mote [26]. Although the system is built using commercially available components, it has a high power draw of 60 mW.

The Polar Heartstrap is a popular wearable heart rate monitor [20]. It consists of a chest strap (transmitter) which includes electrodes and cardiac data acquisition circuitry. The strap wirelessly sends cardiac data measurements to receivers. However, integrated circuits that receive and decode this signal are power hungry and consume around 100 mW. In addition, these monitors only provide the heart rate and not the full electrocardiograms.

6 Conclusions

We present AudioDAQ, a new platform which harvests energy from the microphone bias signal of a mobile phone, provides both a power and communications channel to external sensor peripherals, and enables a new class of sensing applications that require extended periods of data collection. We envision that with the AudioDAQ platform, it will soon be possible to design peripherals for feature phones with a level of functionality that until now has been limited to smartphones. By using the microphone line for power and communication, and using the phone's built in voice memo application for data capture, we dramatically lower the barrier to entry for phone-centric continuous sensor data collection on all phones. AudioDAQ achieves this by using a simple, ubiquitous, and universal hardware interface, and eliminating the custom software development traditionally required to achieve similar levels of functionality. We show the feasibility of our approach with a low-cost, low-power EKG monitor based on the AudioDAQ architecture that captures cardiac signals continuously for extended periods of time, and delivers this data to the cloud for processing, storage, and visualization. In summary, our work shows how to leverage the billions of mobile phones already in existence for low-cost, low-rate analog sensor data capture, storage, and delivery.

7 Acknowledgments

We thank Eric Brewer, Kevin Fu, Ye-Sheng Kuo, Alanson Sample, Thomas Schmid, and Joshua Smith for their many insights and suggestions. We also thank the anonymous reviewers for their valuable feedback. This material is based upon work supported by gifts from Google, Microsoft, Nokia, and Texas Instruments. This material is based upon work supported by the National Science Foundation under Grant #0964120 ("CNS-NeTS"), #1059372 ("CI-ADDO-NEW"), and #1111541 ("CNS-CSR"). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

8 References

[1] `http://www.devtoaster.com/products/rev/index.html`, Apr. 2011.

[2] Android meets Arduino. `http://developer.android.com/guide/topics/usb/adk.html`, Apr. 2011.

[3] iCouchPotatoe. `http://www.hackint0sh.org/f131/35975.html`, Apr. 2011.

[4] Alex Winston, LTD. iData (FSK): Freeing the iPhone and iPod. https://sites.google.com/a/alexwinston.com/www/idata.

[5] Apple, Inc. Apple iPhone TTY adapter. `http://store.apple.com/us/product/MA854G/A`, Apr. 2011.

[6] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory sensing. In *Workshop on World Sensor-Web: Mobile Device Centeric Sensor Networks and Applications*, WSW '06, Dec. 2006.

[7] C. Van Mieghem, M. Sabbe, and D. Knockaert. The clinical value of the EKG in noncardiac conditions. In *Chest 125 (4): 1561-76*, Sept. 2004.

[8] R. Chaudhri, G. Borriello, R. Andesron, S. McGuire, and E. O'Rourke. FoneAstra: Enabling remote monitoring of vaccine cold-chains using commodity mobile phones. In *Proceedings of the First Annual Symposium on Computing for Development*, DEV '10, Dec. 2010.

[9] C&K Components. KSR series subminiature tact switch for SMT, Aug. 2012.

[10] M. Huyett. An AFSK interface for Android™ smartphones. QST Magazine, May 2012.

[11] Kobiconn. 3.5 mm, 4 cond plug. `http://www.mouser.com/catalog/specsheets/KC-300914.pdf`, Aug. 2010.

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

[13] L. Fay, V. Misra, and R. Sarpeshkar. A micropower electrocardiogram amplifier. In *IEEE Transactions, Biomedical Circuits and Systems*, Sept. 2009.

[14] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell. The jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 71–84, 2010.

[15] Maxim Integrated. Icm7555/7556 general purpose timers. `http://datasheets.maximintegrated.com/en/ds/ICM7555-ICM7556.pdf`, Nov. 1992.

[16] Maxim Integrated. High-speed, low-voltage, 0.7 ω CMOS analog switches/multiplexers. `http://datasheets.maximintegrated.com/en/ds/MAX4781-MAX4783.pdf`, Feb. 2005.

[17] R. Media. My TV Remote. `http://www.ryzmedia.com`, Apr. 2011.

[18] P. Dutta, and L. Subramanian. Human-enabled microscopic environmental mobile sensing and feedback. In *Proceedings of the AAAI Spring Symposium on Artificial Intelligence for Development*, AI-D '10, Mar. 2010.

[19] P. Germanakos, C. Mourlas, and G. Samaras. A mobile agent approach for ubiquitous and personalized ehealth information systems. In *Proceedings of the Workshop on Personalization for eHealth of the 10th International Conference on user Modeling*, UM '05, July 2005.

[20] Polar. Polar Wear+Hybrid transmitter. `http://www.polar.fi/en/products/accessories/WearLink_hybrid_transmitter`, Apr. 2011.

[21] B. Priyantha, D. Lymberopoulos, and J. Liu. Enabling energy efficient continuous sensing on mobile phones with littlerock. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN '10, pages 420–421, 2010.

[22] Prological Solutions, LLC. H4i: Homemade for integration. `http://www.progical.com/`, Apr. 2011.

[23] R. Honicky, E. A. Brewer, E. Paulos, and R. White. N-smarts: Networked suite of mobile atmospheric real-time sensors. In *Proceedings of the Second ACM SIGCOMM Workshop on Networked system for developing regions*, NSDR '08, Dec. 2008.

[24] Square. Square card reader. `http://www.squareup.com`, Apr. 2011.

[25] SwitchScience. Audio jack modem for iPhone and Android. `http://www.switch-science.com/trac/wiki/ARMS22-SOFTMODEM-HOWTO`, Apr. 2011.

[26] T. R. F. Fulford-Jones, G.-Y. Wei, and M. Welsh. A portable, low-power, wireless two-lead EKG system. In *Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEMBS '04, Sept. 2004.

[27] Texas Instruments. 4-bit synchronous binary counters. `http://www.ti.com/lit/ds/symlink/sn74lv161a.pdf`, Dec. 2005.

[28] Texas Instruments. Ultra-low Iq, 50 mA LDO linear regulators with power good output in sc70 package. `http://www.ti.com/lit/ds/symlink/tps79718.pdf`, Apr. 2012.

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

[30] V. Fuster (ed.). Hurst's. The Heart, Tenth Edition, New York, NY. Mc-Graw-Hill Medical Publishing, ch. 11, 2011.

[31] Vital Wave Consulting. mHealth for development: The opportunity of mobile technology for healthcare in the developing world. United Nations Foundation, Vodafone Foundation, Feb. 2009.