Hijacking Power and Bandwidth from the Mobile Phone's Audio Interface

Ye-Sheng Kuo, Thomas Schmid, Prabal Dutta
Computer Science & Engineering Divison
University of Michigan
Ann Arbor, MI 48109
{samkuo,thschmid,prabal@eecs.umich.edu

ABSTRACT

We endow the digital mobile phone with an analog interface that can parasitically power external peripherals and transfer data to and from them using the existing headset interface. Our design delivers several milliwatts at 3 V to a load and offers a bidirectional communications channel at a data rate of 300 bps. We demonstrate a simple oscilloscope application using this new functionality.

1. INTRODUCTION

The mobile phone is the most pervasive personal communications and computing platform ever created and yet, among its various analog interfaces, only one is open, standardized, and widely accessible: the headset port. In this paper, we take a closer look at this ubiquitous interface and assess its utility for augmenting the mobile phone with a range of phone-powered peripherals. We show that the mobile phone headset port can be used to efficiently power external peripherals and communicate with them, enabling many new phone-centric applications. But, why use the headset port at all? One reason is that it is an open, simple, and ubiquitous interface with documented electrical and mechanical specifications, as Figure 1 shows. Perhaps even more important, the headset interface is backward- and forward-compatible with most mobile phones in use today, so the mobile phone could form the basis for many health and communications applications in developing regions.

The motivation for such mobile phone peripherals comes from several quarters. First, the emergence of mobile phone accessories like the Square Card Reader [1] suggest that simple peripherals that can leverage the headset port have commercial appeal. Second, researchers focused on developing regions have a need for low-power vital signs devices that could directly interface to a mobile phone and be powered from it. Third, leading academics have argued for reformulating introductory computer science curriculums around the mobile phone, but this paper shows how many important electrical engineering concerns like power, communications, and embedded systems can also be explored in this context. This proposal seeks to enable additional devices like the Square Card Reader, provide a basis for new phone peripherals, and enable greater student engagement by allowing EECS students to tinker with their phones.

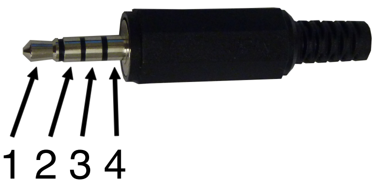


Figure 1: The iPhone headset plug and its pinout. iPhones use a 3.5 mm phono jack/plug to output audio to headphones and receive input from a microphone. The headphone connections are: (1) left earphone (tip), (2) right earphone (ring), (3) common/ground (ring), and (4) microphone (sleeve). The measured impedance between the left (or right) earphone and common is 33 Ω. The measured impedance between the microphone and common is approximately 640 Ω.

This project poses several engineering and research challenges. The output from the audio jack is a low voltage signal, often even lower than typical transistor threshold voltages. To be useful, it must be converted to a higher voltage using energy harvesting and voltage boosting circuits that can operate with input AC voltages in the 200 mV level. Due to the limited voltage headroom, simple rectification is difficult without substantial power losses. It may also require maximum power point tracking. Matching the harvesting circuit's cost, complexity, and conversion efficiency with the ideal audio waveform also presents an iterative co-design problem. Using the audio output to deliver power and data functionality requires exploring several design tradeoffs and concerns.

In this paper, we characterize the power available from the audio jack, design a circuit to harvest this power, and evaluate the efficiency of the conversion. We find that the headset can deliver approximately 16 mW per channel from the iPhone's headset port. We present a circuit that can harvest energy from a single channel and an audio signal that when played on the phone can maximize the output power from the harvesting circuit. We also demonstrate that a pair of (coded) audio signals can be generated by the phone processor and transmitted to both the energy harvesting circuit (power) and a microcontroller (signal) and where the signal can be decoded by the microcontroller. Conversely, we show that the microcontroller can also generate a coded signal that can be read by the mobile phone's microphone input and decoded by the phone to present a stream of digital data. Finally, integrating all of the various pieces, we present a simple oscilloscope application that runs partly on the mobile phone and partly on an external microcontroller powered using the mobile phone's right audio channel. The two processors communicate using the left audio channel (phone to microcontroller) and microphone (microcontroller to phone).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISLPED'10 Design Contest Austin, TX, USA

Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...$10.00.

2. ENERGY HARVESTING

Our first design goal is to harvest energy from the headset jack of a mobile phone, convert it into a more usable form, and achieve high conversion efficiency in the process. We begin by characterizing the AC waveforms that are available, the output impedance of the headset port, and the range of variables available for manipulation. We then design and evaluate an energy harvesting circuit to convert the available power into a more usable form.

2.1 Determining Available Power

We now explore the question of how much power can be harvested from iPhone's headset port. To do so, we use the Faber Acoustical iPhone SignalScope Pro software [2] to generate a range of audio frequencies, from 20 Hz to 24 kHz, and output them over the iPhone's audio port. We find that the output power is independent of frequency, so we use a 5 kHz AC tone in our subsequent experiments. Table 1 shows the settings to generate the output.

Parameter	Value
Hardware	iPhone 3GS [3]
Software	SignalScope Pro [2]
Function	Signal Generator
Output	Headphones
Type	Tone
Frequency	20 Hz to 24 kHz (5 kHz nom)
Amplitude	0.00 dB
Pan	0.000%
Volume	Maximum

Table 1: Experiment parameters for determining the available power from the iPhone 3GS headset port.

A load resistance is connected between the right audio channel and common line on the headset. The load is varied from 0 Ω to 15 kΩ and the output voltage and load current are measured at several points. A linear fit of the data yields the (essentially linear) IV curve shown in Figure 2. From these data, we generate the power transfer curve, which shows that maximum power transfer occurs at 240 mVrms and 66.0 mArms, for a 3.6 Ω load.

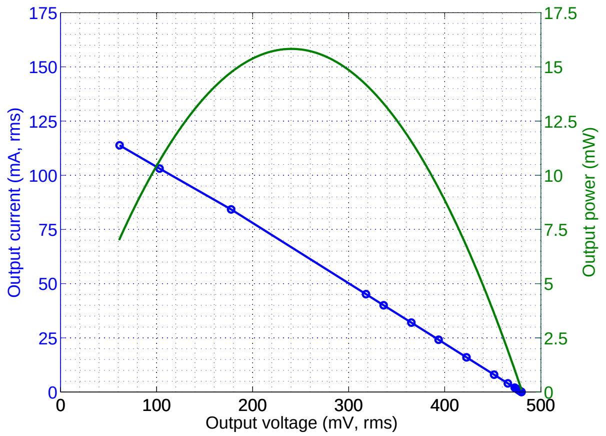
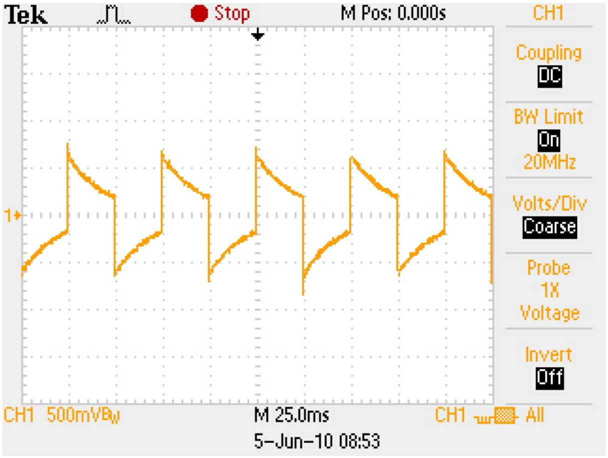


Figure 2: Available power from the iPhone headset jack. These data show that it is possible to draw 15.8 mW from an ideally-matched load of 3.6 Ω, which is enough to power many low-power electronics. To be useful, however, the power must be rectified from AC to DC, boosted to a few volts, and filtered.



Stop
M Pos: 0.000s
CH1
Coupling
DC
BW Limit
On
20MHz
Volts/Div
Coarse
Probe
1X
Voltage
Invert
Off
CH1 500mV BW
M 25.0ms
CH1 All
5-Jun-10 08:53

Figure 3: The audio output waveform observed when a 20 Hz square wave is transmitted over the audio output channel. The sharp transitions followed by characteristic exponential-decay curves show the output is AC-coupled, eliminating the simple option of driving the audio output with a constant DC voltage.

2.2 Design Space Exploration

We next explore the question of how to efficiently harvest the energy produced from the headset output. The two engineering challenges are to increase the signal amplitude and convert the AC signal into a DC one. Figure 2 shows that the open circuit voltage, V_{oc}, is less than 500 mV and that the maximum power point voltage, V_{mpp}, occurs at 240 mV. These voltages are far below the turn on voltages of switching regulators (typically in the range of 800 mV to 900 mV). They are also below the required startup voltage, after rectification, of ultra-low voltage step-up DC-DC converters, like the Seiko S-882Z [4], which require 300 mV to start.

Rectification losses can be significant in both high-power and low-voltage systems. In our case, for example, to achieve maximum power transfer, an RMS current of 66 mA is required. When rectified using even a low-V_f Schottky diode like the DFLS120L, a 200 mV forward voltage drop occurs (See Fig 1 in reference [5]), meaning that 80% of the power is lost during rectification, and only 20% can be delivered to the load.[1] Synchronous rectification is sometimes used to reduce losses, where a FET switch is used instead of a diode [10]. In low-voltage applications like ours, the problem is generating a sufficiently high gate drive voltage to turn on the FET switch. Given the low voltages involved, this would require many stages of (inefficient) voltage multiplication ladders.

We end this section on design alternatives by eliminating two simple, but ultimately unworkable, options: harvesting DC directly from the audio output and harvesting DC from the microphone bias voltage. Figure 3 shows the waveform that is observed on the audio output when a 20 Hz square wave signal is generated on the phone. The characteristic exponential decay curves suggest that the output is AC-coupled, and is therefore a high-pass filter that blocks DC. This eliminates the possibility of simply generating a DC output voltage to power the external devices. Using the microphone bias voltage is also difficult because we plan to use it as the data input channel to the phone, which will be modulated externally.

[1] This assumes that only a single rectifier diode is on the path, which would of course reduce the available power by 50%. If two diodes are on the path, as would be the case for a bridge rectifier, the losses would be substantially higher.

2.3 Harvesting Energy Efficiently

To sidestep the two basic engineering challenges – low-supply voltage and need for rectification – we use a step-up microtransformer, followed by FET-based rectification, followed by (parallel) blocking Schottky diode(s), followed by filter capacitors, as shown in Figure 4. One key element of the design, the microtransformer, leverages a recently introduced device for flyback and step-up for energy harvesting applications. These new transformers are small (6 mm x 6 mm x 3.5 mm), have high coupling coefficients (> 0.95), and are available in a range of turns ratios [6]. We use a 1:20 ratio.

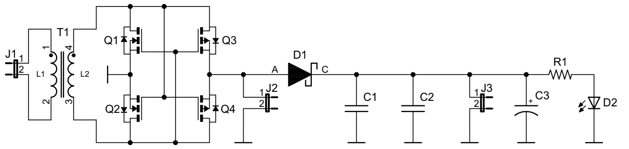


Figure 4: The energy harvesting circuit. A 1:20 microtransformer boosts the input voltage. A FET bridge efficiently rectifies the AC signal to DC. Parallel Schottky diodes provide low-loss blocking to prevent the output filter capacitor from discharging through the FET bridge. An (optional) LED with current-limiting resistor provides a visual power indicator.

The stepped-voltage is passed through a FET bridge for rectification. Since the stepped-up voltage is substantially higher than the FET threshold voltage, the FETs are in conduction and offer marginal loss. Another benefit to stepping-up the voltage is a reduction in current flow through the blocking diode, and therefore a reduction in forward voltage drop. However, since the diode is an exponential device, this unfortunately does not result in a substantial decrease in the forward voltage drop, but it does eliminate the voltage drop from a second diode in the rectifier. And, since the diode forward voltage drop is a small fraction of the rectified voltage, this design incurs a small inefficiency compared to direct rectification of the low-voltage signal.

Matching the load and source impedances is critical to achieving high-efficiency power transfer from a power supply to its load. In this case, the impedance offered by the microtransformer's primary winding should be matched to the iPhone's audio output port's impedance of 3.6 Ω. The transformer's datasheet states that the primary DC resistance is 200 mΩ and primary inductance is 25 μH, which we verified empirically. Since the transformer's DC resistance is small compared to the power supply's output impedance, we focus on the transformer's impedance. The impedance, X_L, offered by an inductor is

$$X_L = j\omega L = j2\pi f L.$$

Rearranging to solve for f, the desired excitation frequency, gives

$$f = \frac{X_L}{2\pi L}.$$

Substituting our measured impedance and inductance values gives

$$f = \frac{3.6\ \Omega}{2\pi \times 25\mu\text{H}} = 22.9\ \text{kHz}.$$

The target excitation frequency sits just at the edge of the what the iPhone is capable of producing. Fortunately, however, we have complete control over the excitation frequency within the audio band, so we can generate a 22 kHz waveform which will achieve near optimal power transfer to the energy harvester circuit.

2.4 Design Evaluation

To evaluate the performance of our design, we implement the energy harvesting circuit, as shown in Figure 5. The circuit only requires a footprint of 1.0" x 0.35" (although the current board includes a 0.15" unused area for manufacturing reasons) and several header export lines. The circuit's small size makes it suitable for embedding inside a headset plug, like the one shown in Figure 1.

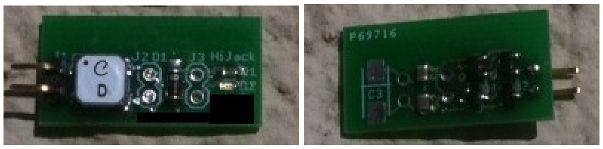

[redacted]

Figure 5: Energy harvesting circuit. The transformer, Schottky diode, LED, and resistor are visible on the top side. The FET rectifier and filter capacitors are visible on bottom side.

Figure 6 shows a trace of the circuit in operation. The iPhone generates a 22 kHz, 500 mV peak-to-peak square wave that is low-pass filtered (substantially lower excitation frequencies result in poor power transfer). The RMS value of the signal is 207 mV, meaning that approximately 15 mW is delivered by the phone, or about 90% of peak power. Channel 1 (orange) shows this filtered audio output signal. Channel 2 (blue) shows a peak 4.24 V signal after rectification using the FET bridge. Channel 3 (magenta) shows the output after passing the rectified signal through (a single) blocking diode, which drops 230 mV at peak current, providing at worst a 94.5% efficiency. Channel 4 (green) shows the voltage across the LED after the signal passes through a high-side current-limiting resistor measured to be 699 Ω.

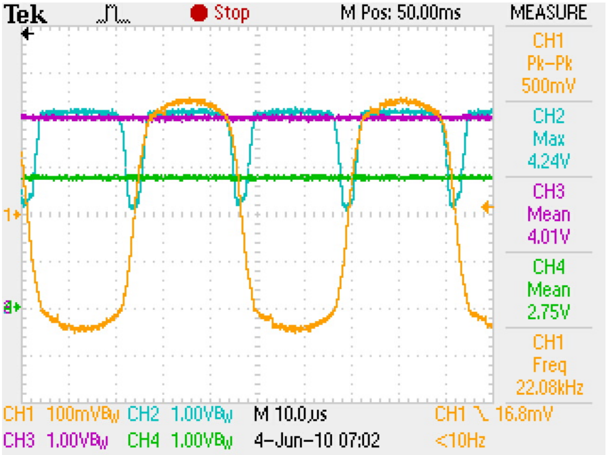


Figure 6: Energy harvesting circuit operation. Channel 1 (orange) shows the filtered audio excitation signal. Channel 2 (blue) shows the signal after rectification. Channel 3 (magenta) shows the output after passing through a blocking diode which drops. Channel 4 (green) shows the voltage across the LED.

The power supply ripple is less than ±10 mV as long as the audio signal is present and the load is static. Our design currently does not include voltage regulation for several reasons: (i) it may not be needed in some applications; (ii) it is not needed for our example application; (iii) the output voltage and power supply buffering is often specific to the particular application. These results show we can harvest energy from the phone's headset port, and convert it into a more useful form, using a simple and inexpensive circuit.

3. DATA COMMUNICATIONS

Our second design goal is to provide bi-directional communications between the mobile phone and a peripheral microcontroller. The requirements for this communication channel are: (i) it must operate in the audio frequency range, and (ii) it must be easy to implement on a microcontroller. The second requirement is necessary as we need to implement both the modulator and demodulator functions inside of a microcontroller since most other integrated circuits that provide a modulation and demodulation functionality are not sufficiently low power, drawing tens of milliwatts [7].

Given these two requirements, we use the well-established Bell 202 signaling technique. Bell 202 employs frequency shift keying (FSK) using two tones. A digital zero, or “space,” is represented by a 1200 Hz tone, while a digital one, or “mark”, is represented by a 2200 Hz tone. Unlike the Bell 202 standard, which specifies 1200 baud communications, we chose to use a lower data rate of 300 baud in order to facilitate an implementation on a low-power microcontroller. At the digital level, we use low-voltage RS-232 signaling to create a virtual universal asynchronous receiver/transmitter (UART) abstraction over the audio serial bit stream. Since the UART protocol adds a start and stop bit to every byte, the effective data rate is 30 bytes/sec.

In order to efficiently implement FSK encoding and decoding on a microcontroller, it is useful to make maximum use of the microcontroller peripherals. W. Lutsch describes one such implementation that uses several hardware features to efficiently modulate and demodulate FSK signals [8]. Figure 7 presents an overview of our implementation of this software-defined radio approach. *Using these hardware accelerators allows for much lower-power operation than would otherwise be possible.*

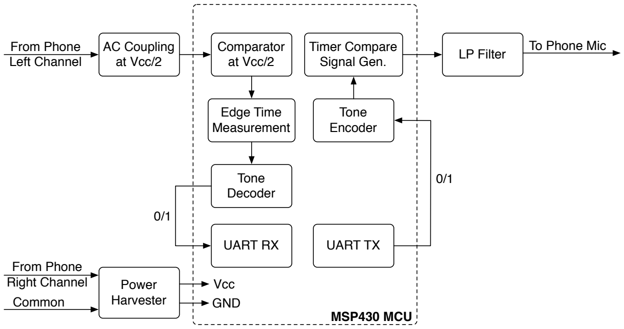


Figure 7: Data communications system architecture. The phone’s headset port exports four wires: right channel, left channel, mic, and common. The right channel provides power to the energy harvesting circuit. The left channel provides data output from the phone to the microcontroller. The mic provides data input from the microcontroller to the phone. Both the phone and the microcontroller implement FSK modulators and demodulators.

The main communication interface for an application running on the microcontroller is the UART receiver and transmitter peripherals. The pins of both the receive and transmit unit of the UART are connected to other microcontroller peripherals for more processing.

On the encoder side, the UART transmitter generates the data bits at 300 baud (Channel 2, blue, on Figure 8). These bits are fed back to the microcontroller into an interrupt line. A timer compare unit generates the correct frequency according to the bit coming from the UART transmitter. The output of the timer compare is a square wave at either 1200 Hz or 2200Hz (Channel 1, yellow, on Figure 8). This signal is sent through a low-pass filter before it is fed into the microphone port of the phone. This filtering reduces the high frequency components of the square wave.

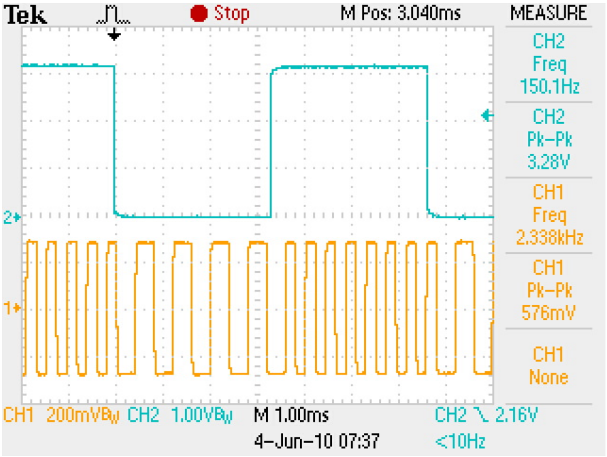
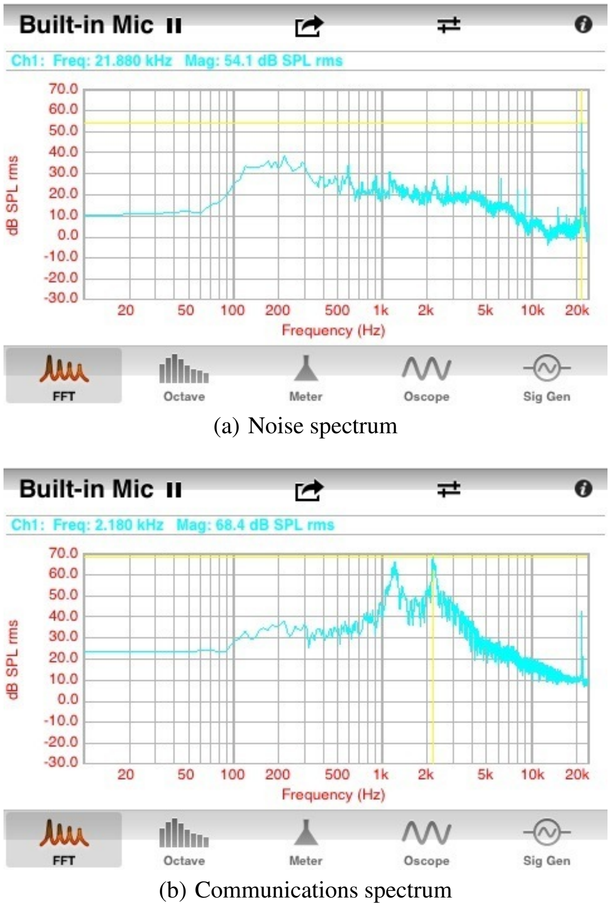


Figure 8: The data stream and its modulated transmission from the microcontroller to the phone. The top trace shows a 300 bps data stream, where a zero-bit is shown from 2.0 ms to 5.3 ms, followed by a one-bit from 5.3 ms to 8.6 ms. The bottom trace shows the FSK-encoded transmission of this data, where the zero-bit is encoded as a 1200 Hz tone and the one-bit is encoded as a 2200 Hz tone. Although the microcontroller-generated signals are square waves, they they do undergo some low-pass filtering.

The left channel headphone output is a ground-referenced AC signal. Unfortunately, we cannot directly process this signal in the microcontroller, and thus we AC couple it to a voltage divider whose midpoint is set to the microcontroller’s Vcc/2. The main idea behind the FSK decoder is to measure the zero-crossing time of the signal, and decide on the frequency. We achieve this by comparing the signal to Vcc/2 in a comparator that is internally connected to a timer capture unit. In software, we calculate the time difference between rising and falling edges using the timer capture interrupt. The result of this decision is then output on a digital IO line that is externally connected back into the receive port of the UART peripheral. The UART receiver is then responsible for decoding the signal at 300 baud.

Implementing the FSK encoder and decoder on the phone is easier as we have a more powerful processor available. This allows us to use more sophisticated algorithms. We implement a non-coherent FSK demodulator on the phone [9]. In non-coherent FSK demodulation, the incoming signal is correlated with four different signals, a pair of sine/cosine signals at 1200 Hz, and a pair of sine/cosine signals at 2200 Hz. The correlation value of the two pairs is sampled and added together. Then, the sums of the two pairs are compared to each other in order to decide which frequency is currently transmitted. A state machine processes the stream of ones and zeros to detect the UART start bit, before it decodes the byte value transmitted in the FSK signal.

The FSK encoder on the phone creates a continuous-phase signal that switches between the two tones according to the bit that is transmitted. Figure 9 shows the audio spectrum used by the phone. Two peaks are clearly visible at 1200 Hz and 2200 Hz, plus the 22 kHz tone used to power the energy harvester. A test of the communication channel from microcontroller to phone found a bit error rate of $0.3 \cdot 10^{-3}$ over a run length of 27,872 bytes.



Ch1: Freq: 21.880 kHz Mag: 54.1 dB SPL rms

(a) Noise spectrum

Ch1: Freq: 2.180 kHz Mag: 68.4 dB SPL rms

(b) Communications spectrum

Figure 9: Audio spectrum utilization. Figure 9(a) shows the noise spectrum of the system (and the 22 kHz power tone). The sound pressure level is less than 40 dB. Figure 9(b) shows the audio spectrum with data communications and power. Two peaks are clearly visible at 1200 Hz and 2200 Hz, as well as the 22 kHz tone used to provide power). We use the FFT function in SignalScope Pro software on the iPhone to show the audio spectrum usage of our system [2].

4. APPLICATIONS

Our final design goal is to show that the various pieces of the system – energy transfer, data input, and data output – can all be combined into a single, integrated, and fully-functional application. For that purpose, we designed a prototype handheld oscilloscope, as shown in Figure 10. This system illustrates a canonical handheld instrument that uses the phone's display for visualization, and the microcontroller's ADC to measure an external signal.

At the heart of this system is the application running on the phone. It generates a 22 kHz tone on the right audio channel to power the microcontroller using the energy harvesting circuit. The left audio channel sends an FSK modulated data stream to the microcontroller. The phone's microphone input receives an FSK-modulated data stream from the microcontroller. The phone application implements a non-coherent FSK demodulator as well as a continuous-phase FSK modulator.

The user interface provides visual feedback to the user of the voltage measured using the microcontroller's ADC input. Figure 11 shows a screenshot close-up of the application running on an iPhone. The graph displays a historic view of measured data, while the text box shows the last received measurement. The data are transmitted from the microcontroller at regular sample intervals. A slider allows the user to change the sampling rate from about 1/10 to 30 Hz, the limit given the 300 bps encoding of the FSK signal. The slider setting is sent over the left channel to the microcontroller, which decodes the signal and changes its sample rate accordingly.

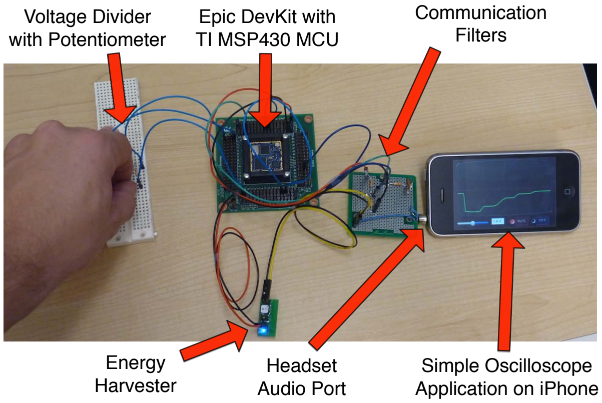
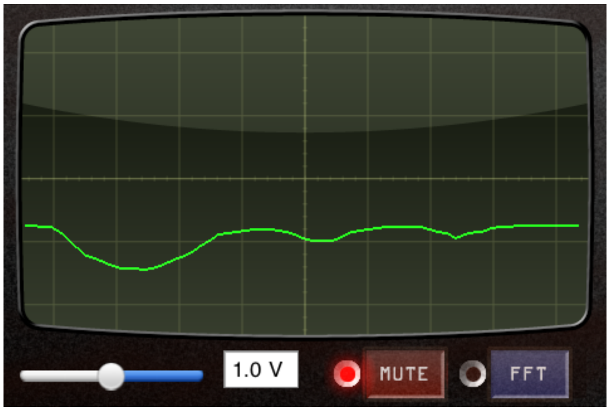


Figure 10: Prototype phone-centric oscilloscope application in full operation. The system consists of four distinct subsystems which are all shown working together: (i) iPhone; (ii) communication filters; (iii) energy harvester; (iv) microcontroller with potentiometer simulating a resistive sensor. The blue LED, located at the bottom of the energy harvester, is turned on and clearly visible in this figure.



1.0 V

Figure 11: Screenshot of the iPhone Oscilloscope application. The slider controls the sampling rate by sending the slider value to the microcontroller over the left audio channel. The sampled ADC value from the microcontroller is sent to the phone over the microphone channel and is visualized on the oscilloscope screen and displayed in a textbox next to the slider. This figure illustrates integrated operation of all four system components.

The microcontroller draws a mere 0.7 mA at 2.8 V while running the full application including FSK encoder and decoder, UART, and ADC to sample the sensor. This is less than 2 mW, leaving significant power for more sophisticated sensing and signal conditioning.

One of the motivations for harvesting energy, rather than directly powering a peripheral with an external battery, is to reduce the form factor of a peripheral. While the prototype is large due to the use of a development kit and protoboard, the active components used could be integrated onto a much smaller circuit board. For example, the two largest components – a transformer and a TI MSP430F1611 microcontroller – measure a mere 6 mm x 6 mm and 9 mm x 9 mm, respectively. In the next version, we plan to use a newer TI MSP430 (<36 mm^2) allowing us to incorporate all the components on a circuit board of about 1.0" x 0.20" size. This is small enough to carry in a pocket along with a phone, allowing any electrical engineering student to carry an oscilloscope or multimeter at all times. Or it would allow the addition of many types of sensors, be it ozone, gas, or even medical (like EKG or blood oxygen) to any phone that has a headset connector and provides rudimentary programming capabilities.

5. CONCLUSION

Battery-free, plug-and-play operation is one reason that USB has been a popular and effective interface on traditional computers, and increasingly on high-end smartphones as well. Although many vendors offer proprietary interfaces (e.g. the iPhone docking connector), the vast majority of mobile phones do not offer a standardized power and analog data interface. In this paper we show that it is possible to augment the ubiquitous headset jack with exactly this functionality.

With the basics of powering external devices and communicating with them now in place, it will soon be possible to plug a small circuit board into almost any existing mobile phone, power an external sensor (e.g. EKG, PulseOx, etc.), bias it with a voltage, read its analog output back, perform some sophisticated filtering, and present an instant result. In short, a parasitically-powered analog expansion interface for the mobile phone will enable many new applications that are either impossible or infeasible with today's technology. The mobile phone has already become *the* personal computer. Tomorrow it could become *the* oscilloscope, volt-ohm-meter, and digital stethoscope as well. This project proposes a small, but important, step toward enabling that future.

Acknowledgments

We would like to thank Eric Brewer for originally suggesting this problem and the anonymous reviewers for their helpful feedback. This material is supported in part by NSF award CNS-0964120, a University of Michigan Research Incentive Award, and a Swiss National Science Foundation Fellowship. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the listed funding agencies.

6. REFERENCES

[1] `http://www.squareup.com`.

[2] `http://www.faberacoustical.com/products/iphone/signalscope_pro/`.

[3] `http://www.apple.com/iphone/iphone-3gs/`.

[4] `http://www.sii-ic.com/en/product1.jsp?subcatID=3&productID=1788`.

[5] `http://www.diodes.com/datasheets/ds30444.pdf`.

[6] `http://www.coilcraft.com/lpr6235.cfm`.

[7] `http://www.onsemi.com/pub_link/Collateral/MC1496-D.PDF`.

[8] W. Lutsch. Msp430 embedded soft-modem demo. `http://focus.ti.com/lit/an/slaa204/slaa204.pdf`, 2004.

[9] J. Proakis and M. Salehi. Communication systems engineering. 2002.

[10] M. Seeman, S. Sanders, and J. Rabaey. An ultra-low-power power management IC for wireless sensor nodes. In *IEEE Custom Integrated Circuits Conference*, pages 567–570, 2007.