PowerPhone: Unleashing the Acoustic Sensing Capability of Smartphones

Shirui Cao*, Dong Li*, Sunghoon Ivan Lee†, Jie Xiong†

ACM MobiCom 2023
Wireless Sensing on Smartphones

Wireless sensing brings more intelligence to smartphones

Acoustic sensing is the *most extensively* studied sensing modality

LiDAR sensing  UWB sensing  Acoustic sensing
Advantage One: Wide Availability on Smartphones

Dedicated modules
- LiDAR sensor
- UWB sensor

Ubiquitous modules
- Speaker
- Microphone
Advantage Two: Fine Sensing Granularity

Acoustic signals enable applications that support fine sensing granularity.

- Finger tracking
- Respiration monitoring
- Eye blink detection

Levels:
- Centimeter level
- Millimeter level
- Submillimeter level
Fundamental Limitation

Acoustic sensing performance is constrained by the limited sampling rate

Human speaking and hearing frequency < 20 kHz

Sampling rates for speaker and microphone 44.1 kHz or 48 kHz
Great Benefits of Higher Sampling Rates

Increasing sampling rate can improve the **sensing resolution**

Sampling rate = **48 kHz**
Two targets are **indistinguishable**

Sampling rate = **192 kHz**
Two targets are **distinguishable**

---

Ground truth ranges

---

Normalized FT

Range (m)

Normalized FT

Range (m)
Great Benefits of Higher Sampling Rates

Increasing sampling rate can boost the **sensing granularity**

Sampling rate = **48 kHz**
- Cannot capture 0.1 mm displacement

Sampling rate = **192 kHz**
- Successfully capture 0.1 mm displacement
Great Benefits of Higher Sampling Rates

Increasing sampling rate can increase the **sensing range**

- **Sampling rate = 48 kHz**
  - Cannot sense a target at 5 m

- **Sampling rate = 192 kHz**
  - Successfully sense the target at 5 m
Opportunities Hidden in Smartphone Hardware

We analyzed more than 100 smartphones and found their hidden capabilities

<table>
<thead>
<tr>
<th>Brand</th>
<th>Lineup</th>
<th># of Models</th>
<th>Audio Codec</th>
<th>RX $f_s$</th>
<th>Speaker Amplifier</th>
<th>TX $f_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google</td>
<td>Pixel 4</td>
<td>1</td>
<td>Knowles IA8505</td>
<td>192 kHz</td>
<td>Cirrus Logic CS35L36</td>
<td>384 kHz</td>
</tr>
<tr>
<td>Google</td>
<td>Pixel 4a/5/5a</td>
<td>3</td>
<td>Realtek RT5514</td>
<td>192 kHz</td>
<td>Cirrus Logic CS35L41</td>
<td>192 kHz</td>
</tr>
<tr>
<td>Samsung</td>
<td>Galaxy S21 lineup (Exynos)</td>
<td>4</td>
<td>Exynos 2100 (integrated)</td>
<td>384 kHz</td>
<td>Cirrus Logic CS35L41</td>
<td>192 kHz</td>
</tr>
<tr>
<td>Samsung</td>
<td>Galaxy A50 lineup</td>
<td>2</td>
<td>Exynos 9610 (integrated)</td>
<td>384 kHz</td>
<td>Cirrus Logic CS35L41</td>
<td>192 kHz</td>
</tr>
<tr>
<td>Samsung</td>
<td>Galaxy A10/A20/A30/A40 lineup</td>
<td>10</td>
<td>Samsung COD3035x</td>
<td>192 kHz</td>
<td>Silicon Mitus SMA1301</td>
<td>192 kHz</td>
</tr>
<tr>
<td>Samsung</td>
<td>Galaxy S9 lineup (Snapdragon)</td>
<td>2</td>
<td>Qualcomm WCD9341</td>
<td>192 kHz</td>
<td>Maxim MAX98512</td>
<td>192 kHz</td>
</tr>
<tr>
<td>Samsung</td>
<td>Galaxy S10 lineup (Exynos)</td>
<td>5</td>
<td>Cirrus Logic CS47L93</td>
<td>192 kHz</td>
<td>Cirrus Logic CS35L40</td>
<td>192 kHz</td>
</tr>
<tr>
<td>Xiaomi</td>
<td>Redmi K40 lineup</td>
<td>4</td>
<td>Qualcomm WCD9380</td>
<td>192 kHz</td>
<td>Cirrus Logic CS35L41</td>
<td>192 kHz</td>
</tr>
<tr>
<td>Xiaomi</td>
<td>Mi 9/10/11/12/13 lineup</td>
<td>24</td>
<td>Qualcomm WCD93xx</td>
<td>192 kHz</td>
<td>Cirrus Logic CS35L41</td>
<td>192 kHz</td>
</tr>
<tr>
<td>Xiaomi</td>
<td>Redmi Note 8/8T/9Pro lineup</td>
<td>3</td>
<td>Qualcomm WCD937x</td>
<td>192 kHz</td>
<td>TI TAS2562</td>
<td>96 kHz</td>
</tr>
<tr>
<td>Huawei</td>
<td>P40 lineup</td>
<td>3</td>
<td>HiSilicon Hi6405</td>
<td>384 kHz</td>
<td>TI TAS2564</td>
<td>192 kHz</td>
</tr>
<tr>
<td>Huawei</td>
<td>Mate20/30/40, P30, Nova6/7 lineup</td>
<td>24</td>
<td>HiSilicon Hi6xxx series</td>
<td>192 kHz</td>
<td>NXP TFA9874</td>
<td>96 kHz</td>
</tr>
<tr>
<td>ASUS</td>
<td>Zenfone 9</td>
<td>1</td>
<td>Qualcomm WCD9380</td>
<td>192 kHz</td>
<td>Qualcomm WSA8835</td>
<td>384 kHz</td>
</tr>
<tr>
<td>Motorola</td>
<td>Edge (2020)</td>
<td>1</td>
<td>Cirrus Logic CS47L35</td>
<td>192 kHz</td>
<td>Cirrus Logic CS35L41</td>
<td>192 kHz</td>
</tr>
<tr>
<td>Sony</td>
<td>Xperia 1/5 II/III, Xperia 10 lineup</td>
<td>6</td>
<td>Qualcomm WCD93xx</td>
<td>192 kHz</td>
<td>Cirrus Logic CS35L41</td>
<td>192 kHz</td>
</tr>
<tr>
<td>Vivo</td>
<td>iQOO Neo3 5G</td>
<td>1</td>
<td>Qualcomm WCD9341</td>
<td>192 kHz</td>
<td>Maxim MAX98928</td>
<td>48 kHz</td>
</tr>
<tr>
<td>OnePlus</td>
<td>OnePlus 6 lineup</td>
<td>2</td>
<td>Qualcomm WCD9341</td>
<td>192 kHz</td>
<td>Maxim MAX98928</td>
<td>48 kHz</td>
</tr>
<tr>
<td>OnePlus</td>
<td>OnePlus 9 lineup</td>
<td>3</td>
<td>Qualcomm WCD9385</td>
<td>192 kHz</td>
<td>NXP TFA98xx series</td>
<td>48 kHz</td>
</tr>
<tr>
<td>Oppo</td>
<td>Realme X50 lineup</td>
<td>4</td>
<td>Qualcomm WCD9385</td>
<td>192 kHz</td>
<td>NXP TFA98xx series</td>
<td>48 kHz</td>
</tr>
</tbody>
</table>
Can we break the audio sampling rate limits of a smartphone, to unleash its hidden acoustic sensing capability?
Introducing PowerPhone

- A pure software reconfiguration
- Support higher sampling rates (E.g., 192 kHz)
- No extra hardware required
First Challenge: Black Box Hardware Structure
Hardware Fundamentals

• Electric schematics are proprietary, but the Linux Kernel are open-source

• We reverse-engineered smartphones through the Device-Tree-Overlays (DTOs) in their Linux kernel

• DTO allows us to know the hardware topology inside the smartphones without physically teardown
Hardware Fundamentals

- We summarize an audio hardware topology based on 100+ smartphones
Driver Reconfiguration

- Configure the Phase-Lock Loop (PLL) to generate correct leading bit-clock
- Modify the registers inside codec and amplifier through SPI or I2C to support 192 kHz sampling rates
- Change Direct-Memory-Access (DMA) bit-rate and memory mapping size
Second Challenge:
Sampling Rates Remain at 48 kHz
Analysis

Multi-layer Signal Chain

- Android OS has many layers in its audio chain
- We summarized them into four layers
  - Linux Kernel Driver Layer
  - Hardware Abstraction Layer
  - System Service Layer
  - User Layer (Public API)
- We need to figure out which layer down-sampled the audio stream
Our Solution: Layer-By-Layer Analysis

• Find out who down-sampled the audio stream is not trivial, especially the HAL is in binary form

• To access the output of the Linux Kernel Driver, we ported Ubuntu Touch OS to the smartphone

• To access the output of the HAL and System Service Layer, we compiled and built our own Android OS based on LineageOS
Our Findings

• We found that **System Service Layer** down-sampled the audio

• We fixed such behavior by changing hard-coded configuration files
Frequency Responses

![Frequency Responses Graph](image_url)

- FFT Amplitude (dBFS)
- Frequency (Hz)

- **Motorola Edge (2020)**
- **Xiaomi Redmi Note 9 Pro**
- **Samsung Galaxy S10 (Exynos)**
- **Samsung Galaxy S9+ (Snapdragon)**
Chirp Signals

![Diagram of chirp signals with smartphone and target illustration]
Resolution Benchmark

Experiment Setup

Results

Graph showing resolvability vs. distance between two targets for different settings and devices.
Resolution Field Study

Gestures

Factory Settings (48 kHz)

PowerPhone (192 kHz)
New Application: Wireless Charger Leakages

- Leakages from capacitors and inductors
New Application: Results

- Classification using Vision Transformer (ViT) network
- Overall accuracy: 96.2%
Conclusion

• Reconfigured smartphone sampling rates to 192kHz and open-sourced implementation
• Enhanced sensing resolution (1 cm), granularity (2 um), and range (6 m)
• Enabling many new ultrasonic sensing applications, e.g., home appliance monitoring
Check Our Website!

- Step-by-step instruction, source codes, system images, etc.
- Future updates / new supported smartphones.
- https://powerphone.github.io