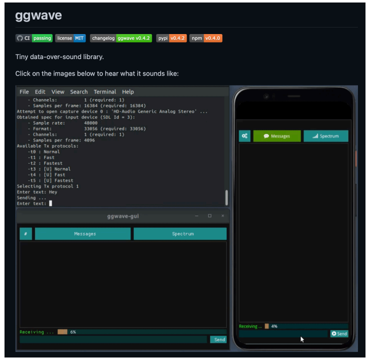
Data over Sound Technology & Audio Processing on ARM/RISC-V

Alastair Paragas
PHYS536 – Introduction to Acoustics

BE BOUNDLESS

Data over Sound


ggwave

Tiny data-over-sound library.

Click on the images below to hear what it sounds like:

```
    - Channels:          1 (required: 1)
    - Samples per frame: 16384 (required: 16384)
Attempt to open capture device 0 : 'HD-Audio Generic Analog Stereo' ...
Obtained spec for input device (SDL Id = 3):
    - Sample rate:       48000
    - Format:            33056 (required: 33056)
    - Channels:          1 (required: 1)
    - Samples per frame: 4096
Available Tx protocols:
    -t0 : Normal
    -t1 : Fast
    -t2 : Fastest
    -t3 : [U] Normal
    -t4 : [U] Fast
    -t5 : [U] Fastest
Selecting Tx protocol 1
Enter text: Hey
Sending ...
Enter text: 
```

Chirp: SDK & (defunct) company
Operates at 18-20 kHz spectrum (most smartphones operate 20Hz-20kHz). Transmission includes: Chirp start signal bits + payload length + actual payload data + CRC error detection code + Reed-Solomon error correction code. Incubated 2011, launched a UK company in 2012 and acquired by Sonos in 2020.

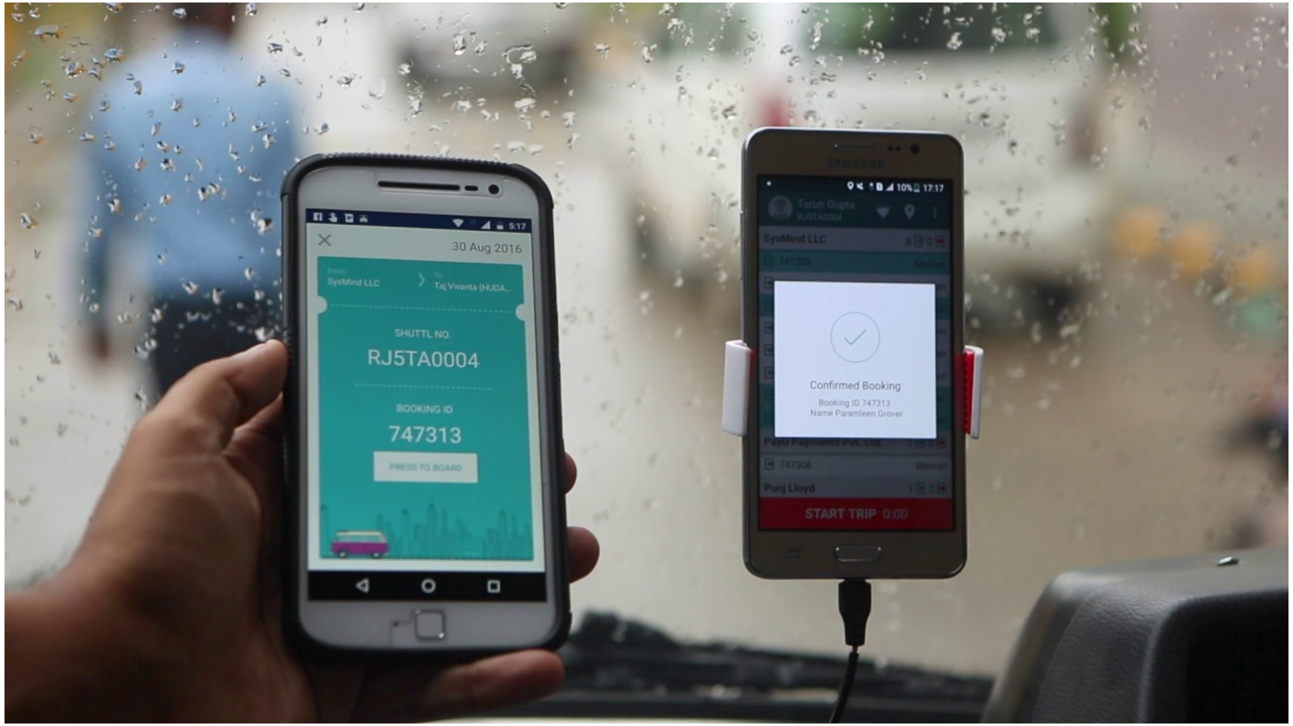
- **Shuttl**: Bus aggregator company in India where Chirp is used for boarding process; customer's phone app generates an audio used by driver's phone app to confirm booking/boarding.
- **EDF Heysham 1 Nuclear Power Plant:** Transmit sensor reading data (50 meters over 12 hours) RF ban due to fear of interference with older tech predating RF legislation & standard shielded cables too bulky/costly.

Ggwave: Open Source Project
Each time slice sends 3 bytes of data. Split 3 bytes into 6 4-bit chunks. 16 possible values per chunk * 6 chunks = 96 values represented as distinct equally-spaced frequencies over a 4.5kHz band that can start at 1.875kHz or 15k Hz

Google Tone
Google Chrome Browser extension to transmit URLs using a computer's speaker (transmitting) and microphone (receiving)

UNIVERSITY *of* WASHINGTON

Data over Sound



UNIVERSITY *of* WASHINGTON

Comparing communication protocols

Features	Chirp	QR	NFC	Bluetooth	Wifi
Max Data Rate	1 kilobits/s	3kb/image	424 kilobits/s	2 megabits/s (Latest spec: Bluetooth 5.3)	2.4 gigabits/s (Latest spec: Wifi 6)
Typical Max Range	100 meters	1m image size per 10 m distance	10 cm (ISO 15693 can extend this to 1m)	Class 1: 100 meters (industrial) Class 2: 10 meters (Bluetooth 5 spec increased possible range to 243 meters)	Most popular frequencies: 2.4GHz @45 meters 5Ghz @15 meters
2-way communication	✓		✓	✓	✓
1-to-many broadcast	✓				
Can operate without line of sight	✓			✓	✓
Frequency Spectrum	20Hz-20kHz (most speakers/ microphones designed for this spectrum; 17-20kHz better for human imperceptability)	N/A, visual	13.56MHz	2.45GHz	Most popular frequencies: 2.4GHz (longer range, worse perf) & 5GHz (shorter range, better perf)

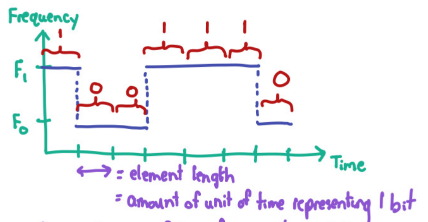
Comparing communication protocols

Features	Chirp	QR	NFC	Bluetooth	Wifi
Can operate without setup	✅	✅	✅		
Can operate with minimal power	✅	✅	✅	✅	
Can be confined within a room/ walls	✅	✅	✅		
Can transmit with cheap electronics	✅	✅	✅	✅	
Can receive with cheap electronics	✅		✅		
Can operate with RF restrictions	✅	✅	✅		

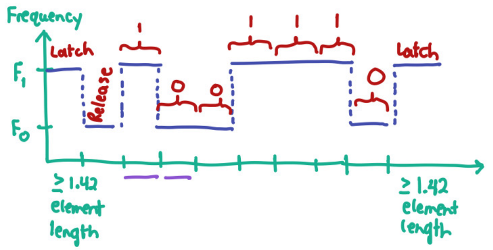
Modulation: digital to analog transmission

Frequency Shift Keying

Say we want to send binary data 1001110.
0 represented by frequency F_0, 1 represented by frequency F_1

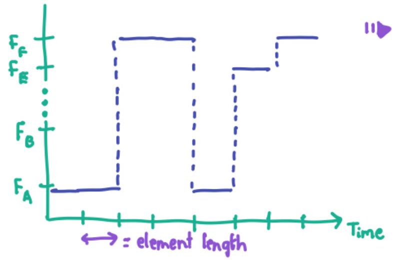


In synchronous FSK, reference clock determines the unit of time used for decoding a frequency signa into a bit



In asynchronous FSK, bit patterns control timing during decoding

What about instead of representing data in terms of 2 states (0,1), represent it instead with M states?
Let possible states = {A, B, C, D, E, F}
We want to send AAFFAEF



→ Ggwave uses M-ary FSK (also Chirp!)
Chunking bits together to send maximum # of bits over a limited number of frequencies

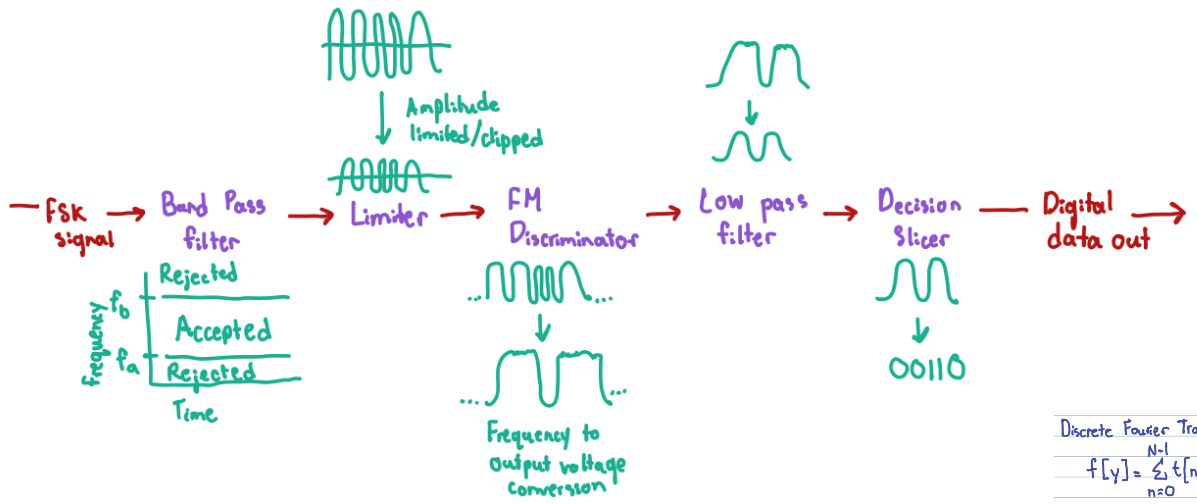
4 bits/chunk * 6 chunks at any given time
Each chunk represented by $2^4 = 16$ frequencies
So, 96 total frequencies used at any given time

Notice if: ggwave used 5-bit chunks instead of 4-bit chunks (and given <= 96 frequencies constraint), 96/2^5 = 3, so we can only send 3 5-bit chunks instead of 6 4-bit chunks. Smaller chunks while being able to send more data, might require more complex demodulators which might not be able to process and reconstruct sent data as quickly.

With 5-bit chunks, 15 total bits sent each time
With 4-bit chunks, 24 total bits sent each time (more bits sent with same amount of frequencies)

UNIVERSITY *of* WASHINGTON

Demodulation: analog transmission to digital



Discrete Fourier Transform:

$$f[y] = \sum_{n=0}^{N-1} t[n] e^{-i2\pi k \frac{n}{N}}$$

$f = \langle f[0], f[1], \dots, f[y], \dots \rangle$ ⇒ A vector/array where each element represents amplitude of some frequency

$t = \langle t[0], t[1], \dots, t[n], \dots, t[N-1] \rangle$ ⇒ A vector/array where each element represents frequency at some moment in time

$\text{eulerValues} = \langle 1, e^{-i2\pi k(\frac{1}{N})}, \dots, e^{-i2\pi k(\frac{N-1}{N})} \rangle$ ⇒ A vector/array of same size as N

Nyquist theorem: Given maximum frequency f Hz (cycles/sec), we want enough sample points N = 2f per second

UNIVERSITY of WASHINGTON

x86, ARM & RISC-V processors

Intel-originated x86 processor architecture & AMD x64: 64-bit extension to x86

- Intel owns original x86 architecture patent & AMD owns the 64-bit extension patent to x86. They cross-license to each other (duopoly)
- x86/x64 - a complex instruction set architecture: instruction/opcode can take multiple cycles to execute. Architecture with many complex opcodes that are more analogous to higher-level languages. Born out of time when memory was expensive and compilers were simple.

RISC-V open-source processor architecture

- Open-source consortium of processor architecture/blueprints: Google (US), Qualcomm (US), SiFive (US), Syntacore (owned by Yadro, Russia), ICT (China), Andes Technology (Taiwan).
- Published 2 ISA specs: 2019 & 2021. Much simpler ISAs than ARM.
- Moved from being headquartered in US to Switzerland in 2019. Permits unrestricted use of ISA for hardware/ software design. Cue chip war & CHIPS and Science Act legislation

ARM processor architecture

- ARM licenses processor architecture/ISA licenses as well as modification rights. They however, do not manufacture actual processor hardware. Customers: Nvidia (Nvidia Grace CPU), Apple (M1, M2), Qualcomm/Nuvia (Oryon CPU), etc
- RISC: instruction/opcode takes 1 cycle to execute. Preference for much less and much simpler opcodes. With cheaper memory and advancement of compilers, RISC architectures unburdened of supporting complex operations, can focus on more registers, larger cache and things like performance per watt.

UNIVERSITY of WASHINGTON

Digital Signal Processing on x86

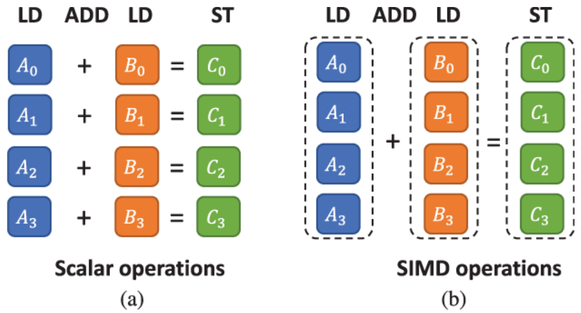
SIMD: Processing Vectors of ints/real numbers & Smart Loops

Digital Signal Processing algorithms often work with multiple chunks of transmitted for a given timestep (for ex: ggwave's 6 4-bit chunks sent out across 96 known frequencies. We can do parallel processing across 96 frequencies at a given processor clock cycle (timestep)! These processors can consume 100-500 watts.

MMX (1996, Intel) & 3DNow! (1998, AMD): Can pack either 2 32-bit, 4 16-bit or 8 8-bit integers into 1 of 8 floating point units (same units used for floating-point calculations) of an Intel Processor. As such, using MMX for integer-valued vector operations prevented processing of real numbers at the same time and also cleared FPU. AMD's 3DNow! can deal with real-valued vectors.

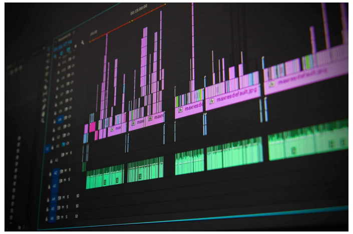
SSE (1999, Intel): SSE1 Can pack 4 32-bit floats into 1 of 16 dedicated registers. SSE2 (2001) can pack either 4 32-bit, 2 64-bit floats, 16 8-bit integers, 8 16-bit integers, 4 32-bit integers or 2 64-bit integers. SSE3 (2006) had more instructions and support for operations between values in same registers. SSE4.2 (2008) brought more instructions and by end of 2008/2009, Intel/AMD CPU SSE hardware handled MMX operations.

AVX (2008, Intel): Can pack either 8 32-bit or 4 64-bit floats into 1 of 16 dedicated registers. AVX can also work with operations between 3 vectors at the same time. AVX2 (2013) brought fused multiple add: one opcode to calculate operation a + (b x c) given real-valued vectors a, b and c. AMD's added improvement: not replacing one of 3 registers with the result, storing it into a 4th. However, AVX has to downthrottle CPU speed to 80% because of overheating issues!



Audio Compression on Desktop: 160kbps, 128 kbps web on Free 320kbps, 256kbps on Premium

Audio Editing



UNIVERSITY *of* WASHINGTON

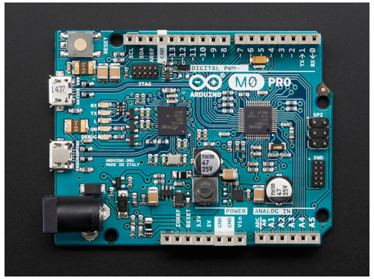
Digital Signal Processing on ARM/RISC-V

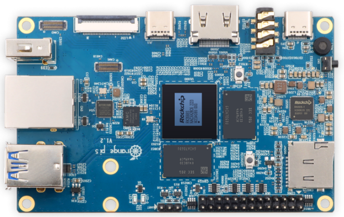
ARM Cortex M for Microcontrollers, ARM Cortex A for mobile
In general, ARM with its RISC architecture targets performance/watt and prefers lower energy consumption than raw performance. Also with simpler overall architecture than x86 and licensed patent rights allows other companies to make modifications (design their own cores/submodules of the processor).

ARM Cortex-M55 (with Helium DSP extension announced 2019): Much like x86 MMX, no dedicated registers for SIMD - floating point unit 128-bit registers hardware are reused for SIMD instructions. Support for operating on 8-bit integer/fixed point, 16-bit integer/fixed-point, 32-bit integer/fixed-point, 16-bit half-precision floating point and 32-bit single-precision floating point. 1 clock cycle to load first 64-bit half of 128-bit vector. On next clock cycle, operation on this first 64-bit half with another register (like MAC - multiply/accumulate: vector dot product) can be done. Interleaved memory access.

ETH Zurich and University of Bologna, using open-source RISC-V architecture, has developed PULP (parallel ultra low power) Digital Signal Processing extensions on the RV32IMC - a single-core 32-bit RISCV architecture.

Andes Technology has developed advanced 64-bit processors with multi-core and DSP capabilities. It comes with 130 DSP instructions. Support for 31-bit/15-bit/7-bit fractional (-1<x<1) numbers, 32-bit/16-bit/8-bit integers as well as 16-bit/8-bit SIMD instructions


Arduino M0 using ARM Atmel Cortex M
0.2 to 1 watt power consumption


Orange Pi 5 with 8 64-bit cores
4x ARM Cortex A55 @1.8GHz (energy cores)
4x ARM Cortex A76 @2.4GHz (perf cores)

UNIVERSITY of WASHINGTON

References

1. "A Beginner's Guide to Digital Signal Processing (DSP)." *A Beginner's Guide to Digital Signal Processing (DSP) | Design Center | Analog Devices*, https://www.analog.com/en/design-center/landing-pages/001/beginners-guide-to-dsp.html.
2. "Bela + Chirp." *Blog.bela.io*, 17 May 2019, https://blog.bela.io/bela-and-chirp-data-over-sound/.
3. *Blending DSP and ML Features into a Low Power General Purpose Processor*. https://armkeil.blob.core.windows.net/developer/Files/pdf/white-paper/blending-dsp-and-ml-features-into-a-low-power-general-purpose-processor.pdf.
4. Carey, Scott. "How EDF Is Using Audio to Monitor Its Nuclear Power Plant Equipment." *Computerworld*, Computerworld, 6 Feb. 2018, https://www.computerworld.com/article/3427522/how-edf-is-using-audio-to-monitor-its-nuclear-power-plant-equipment.html.
5. Cawfree. "Cawfree/Openchirp: A Reverse-Engineered Implementation of the Chirp Data-over-Audio Protocol." *GitHub*, https://github.com/cawfree/OpenChirp.
6. Chattanooga, EPB. "How Far Will Your Wi-Fi Signal Reach?" *EPB*, 19 Jan. 2023, https://epb.com/get-connected/gig-internet/how-far-will-your-wi-fi-signal-reach/.
7. "Chirp Brings Data-over-Sound Capabilities Your Arduino Projects." *Arduino Blog*, 12 Aug. 2019, https://blog.arduino.cc/2019/08/12/chirp-brings-data-over-sound-capabilities-your-arduino-projects/.
8. "Chirping Communication - Sending Data over Sound." *Scientia.global*, 30 June 2022, https://www.scientia.global/wp-content/uploads/2017/10/Chirp.pdf.
9. Corfield, Gareth. "Nuclear Power Station Sensors Are Literally Shouting Their Readings at Each Other." *The Register® - Biting the Hand That Feeds IT*, The Register, 24 Jan. 2017, https://www.theregister.com/2017/01/24/chirp_nuclear_power_station_iot_audio_sensors/.
10. "Data over Sound." *Beeping*, https://beeping.io/.
11. Evanson, Nick. "Explainer: What Are MMX, SSE, and Avx?" *TechSpot*, TechSpot, 28 Dec. 2020, https://www.techspot.com/article/2166-mmx-sse-avx-explained/.
12. "Free Spotify vs Spotify Premium." *SoundGuys*, 2 Jan. 2023, https://www.soundguys.com/free-spotify-vs-spotify-premium-36632/.
13. *FSK: Signals and Demodulation - Rochester Institute of Technology*. http://edge.rit.edu/content/P09141/public/FSK.pdf.
14. Ggerganov. "GGERGANOV/Ggwave: Tiny Data-over-Sound Library." *GitHub*, https://github.com/ggerganov/ggwave.
15. Humphries, Matthew. "Following US Sanctions, China Decides Its Future Lies with RISC Chips." *PCMAG*, PCMag, 2 Dec. 2022, https://www.pcmag.com/news/following-us-sanctions-china-decides-its-future-lies-with-risc-chips.

References

16. *Near Field Communication (NFC) Technology and Measurements White Paper*. https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_application/application_notes/1ma182/1MA182_5E_NFC_WHITE_PAPER.pdf.

17. Orlowski, Andrew. "Chirp! Let's Hear It for Data over Audio." *The Register® - Biting the Hand That Feeds IT*, The Register, 19 Jan. 2017, https://www.theregister.com/2016/11/10/chirp_lets_hear_it_for_data_over_audio/.

18. "RISC vs CISC." *RISC vs. CISC*, https://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/risccisc/.

19. Schiavone, Pasquale, et al. "DSP ISA Extensions for an Open-Source RISC-V Implementation." *4th RISC-V Workshop Proceedings*, RISC-V, 1 Jan. 1970, https://www.research-collection.ethz.ch/handle/20.500.11850/124700.

20. "RISCV Lecture 2021." *Prof Adam Teman*, https://www.eng.biu.ac.il/temanad/files/2021/04/Lecture-3-RISC-V-2021.pdf.

21. Stalker, Ian, and Alan Baldus. "Latest Intel Processors Advance Embedded DSP and SBC System Design." *VITA Technologies*, 5 Apr. 2011, http://vita.mil-embedded.com/articles/latest-dsp-sbc-system-design/.

22. Team, The Arduino. "Arduino M0: Arduino Documentation." *Arduino Documentation | Arduino Documentation*, https://docs.arduino.cc/retired/boards/arduino-m0.

23. *What Is Bluetooth Range? What You Need to Know - 42west, Adorama …* https://www.adorama.com/alc/bluetooth-range/.

24. "What Is SIMD in Digital Signal Processing?" *WolfSound*, https://thewolfsound.com/simd-in-dsp/.

25. Ünsalan, Cem, et al. "2." *Digital Signal Processing Using ARM Cortex-M Based Microcontrollers: Theory and Practice*, Arm Education Media, Cambridge, 2018.