Intercepting, modifying, and generating wireless signals with SDR

Caleb Madrigal
(Public) handle: metem
Website: http://calebmadrigal.com/
Twitter: @caleb_madrigal
Ham call sign: w0hak

About Caleb:

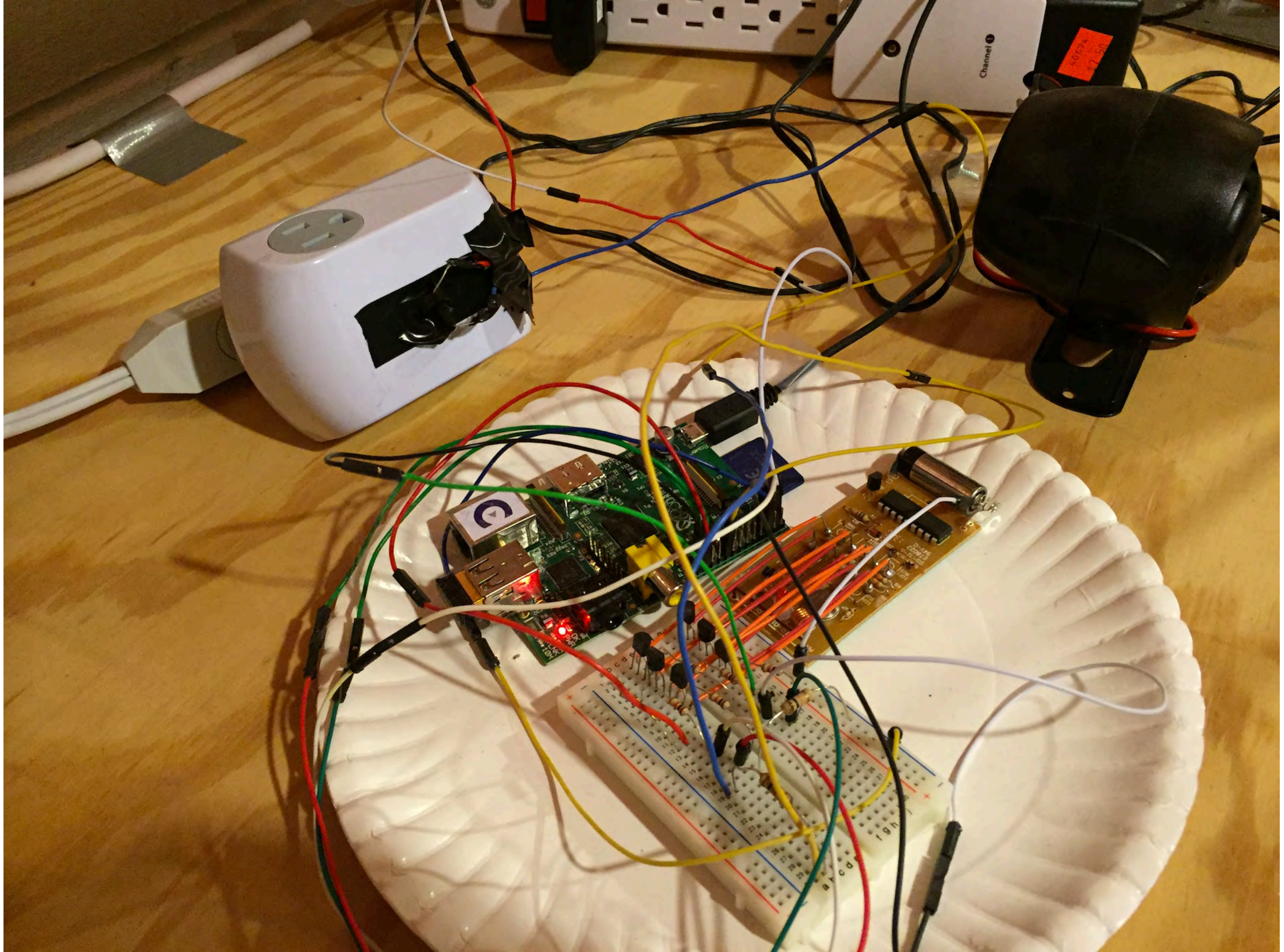
- Programming for about 18 years.
- I most enjoy hacking and mathy stuff (Signal processing, Machine Learning, AI)
    - In general, I find it interesting to hear the "unhearable", see the "unseeable" - to tune in to the subtleties around us.
    - In the computer world, everything is black and white, but the real world is fuzzy and colorful.
- Ontologically, I think the best way to describe myself is as a "Christian Mystic".
- I also love Art - Literature, Music, Artsy movies, etc. I'm also an occasional and unaccomplished poet.

Possible titles for this talk:

- Intercepting, modifying, and generating wireless signals with SDR
- How digital data is transmitted wirelessly
- Controlling wireless IoT devices via crafted radio signals
- How the OSI Physical layer works and how to attack it

Background on what led me to doing this stuff - came in from 3 directions:

- IoT
- Music theory -> Sound analysis
- Wireless hacking


Channel 1

Woods

66 ft
20,1 m
RANGE
RAYON
RANGO

WIRELESS REMOTE 3-PACK
TÉLÉCOMMANDE SANS FIL – PAQUET DE 3
CONTROL REMOTO INALÁMBRICO – PAQUETE DE 3

indoor • intérieur • interior

CH A

1 2 3

CH A

CH A

CH A

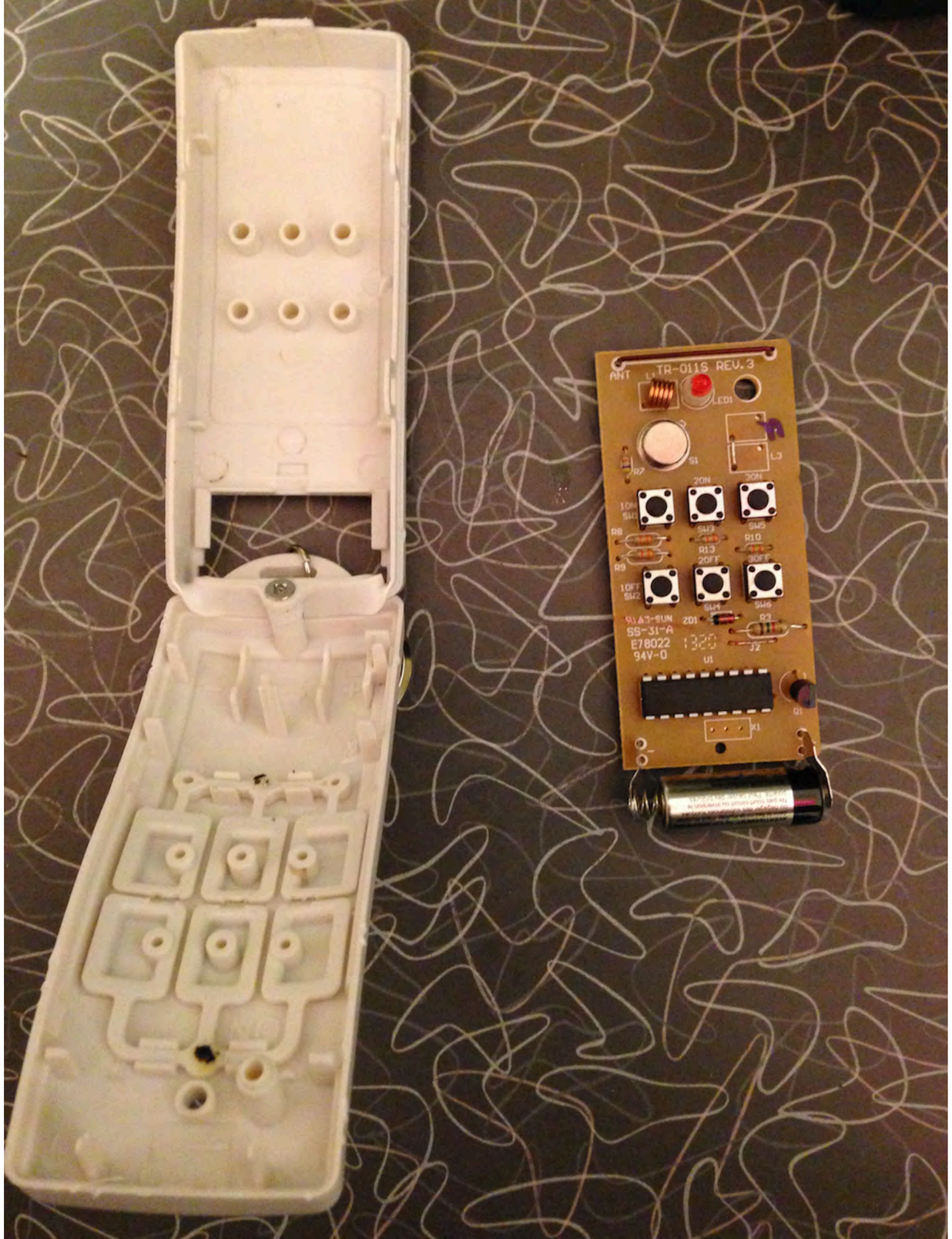
Operate up to 3 different home appliances from across the room, using one remote control

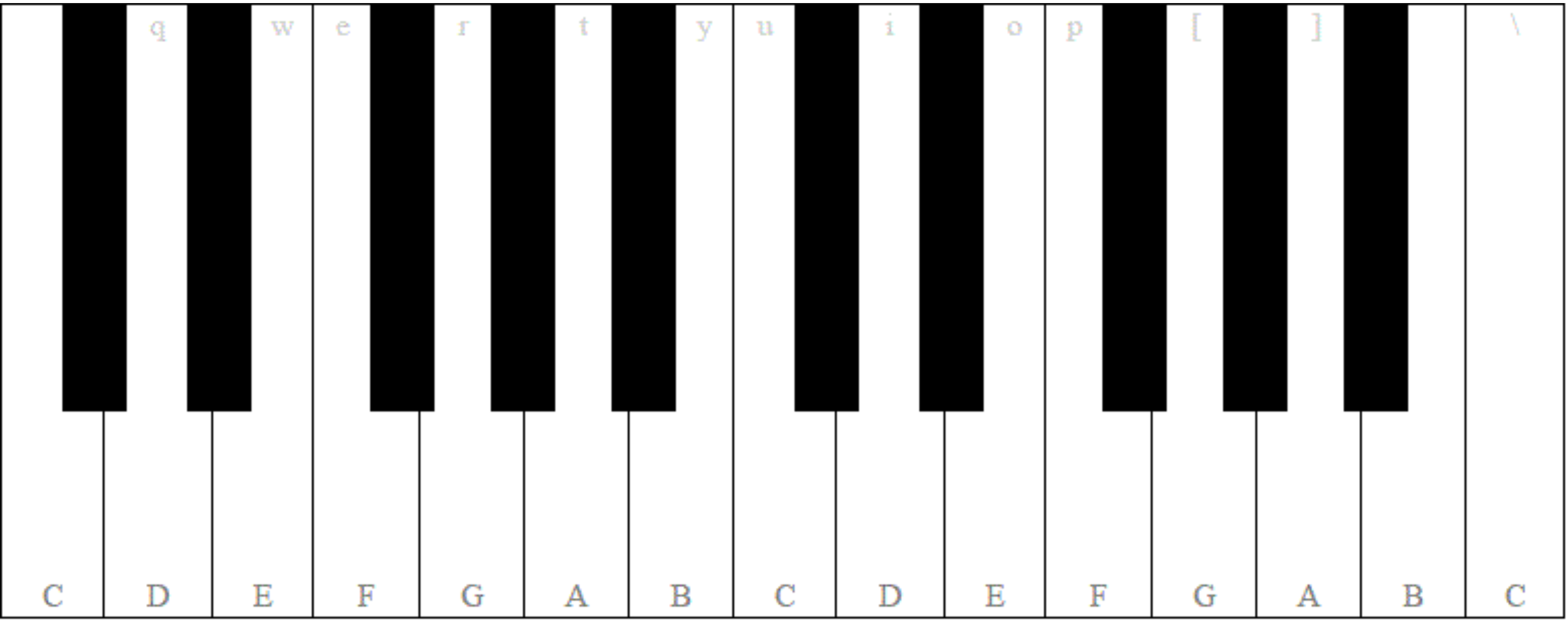
Faites fonctionner jusqu'à 3 différents appareils électriques résidentiels de l'autre côté de la pièce en utilisant une seule télécommande

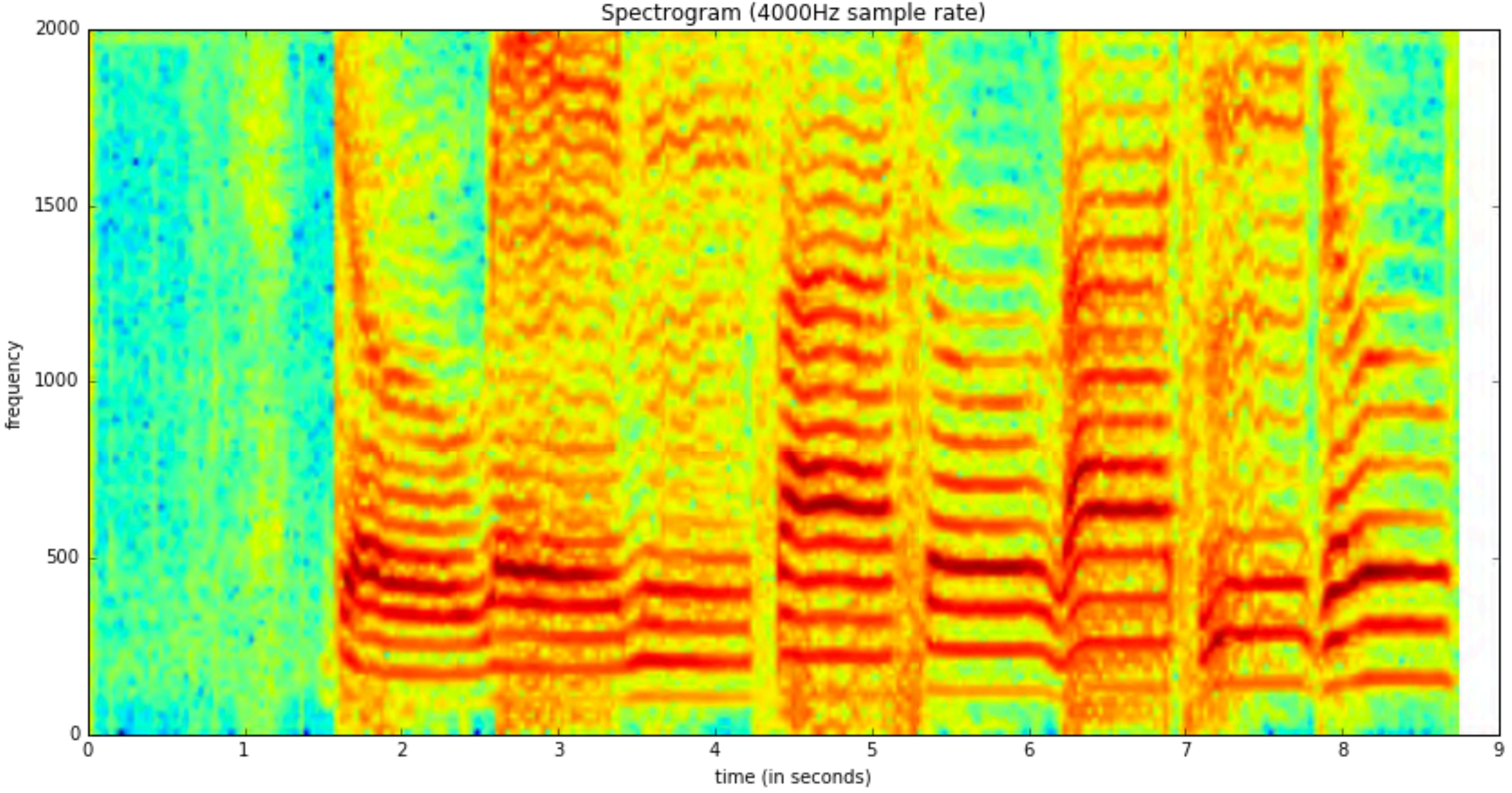
Opere hasta 3 electrodomésticos diferentes desde el otro extremo de la habitación, usando un sólo control remoto

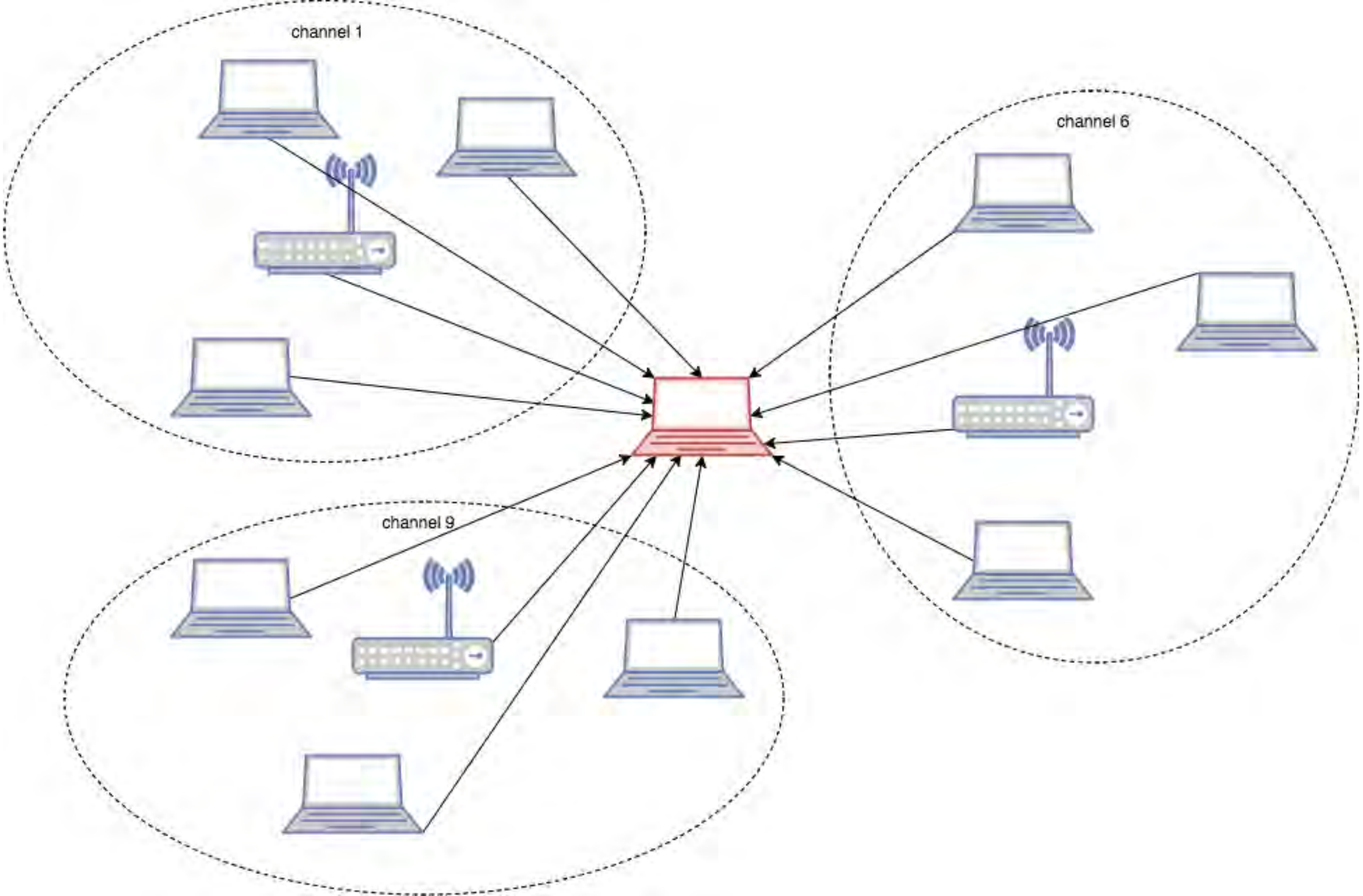
13569

Space saving outlet
Prise compacte
Tomacorriente que ahorra espacio


TR-011S REV.3
ANT
LED1
S1
L3
20N
30N
SW3
SW5
R8
R10
R13
20FF
30FF
SW4
SW6
SS-31-A
E78022
94V-0
U1
X1


q w e r t y u i o p [ ] \
C D E F G A B C D E F G A B C


Spectrogram (4000Hz sample rate)
2000
1500
1000
500
0
frequency
0
1
2
3
4
5
6
7
8
9
time (in seconds)


channel 1
channel 6
channel 9

PACKAGE INDEX
Browse packages
Package submission
List trove classifiers
RSS (latest 40 updates)
RSS (newest 40 packages)
PyPI Tutorial
PyPI Security
PyPI Support
PyPI Bug Reports
PyPI Discussion
PyPI Developer Info

ABOUT
NEWS
DOCUMENTATION
DOWNLOAD
COMMUNITY
FOUNDATION
CORE DEVELOPMENT

trackerjacker 0.7.2

Finds and tracks wifi devices through raw 802.11 monitoring

Download
trackerjacker-0.7.2.tar.gz

Finds and tracks wifi devices through raw 802.11 monitoring.

Not Logged In
Login
Register
Lost Login?
Use OpenID
Login with Google

Status
Nothing to report

Install

```
pip3 install trackerjacker
```

Usage

Find detailed usage like this:

```
trackerjacker -h
```

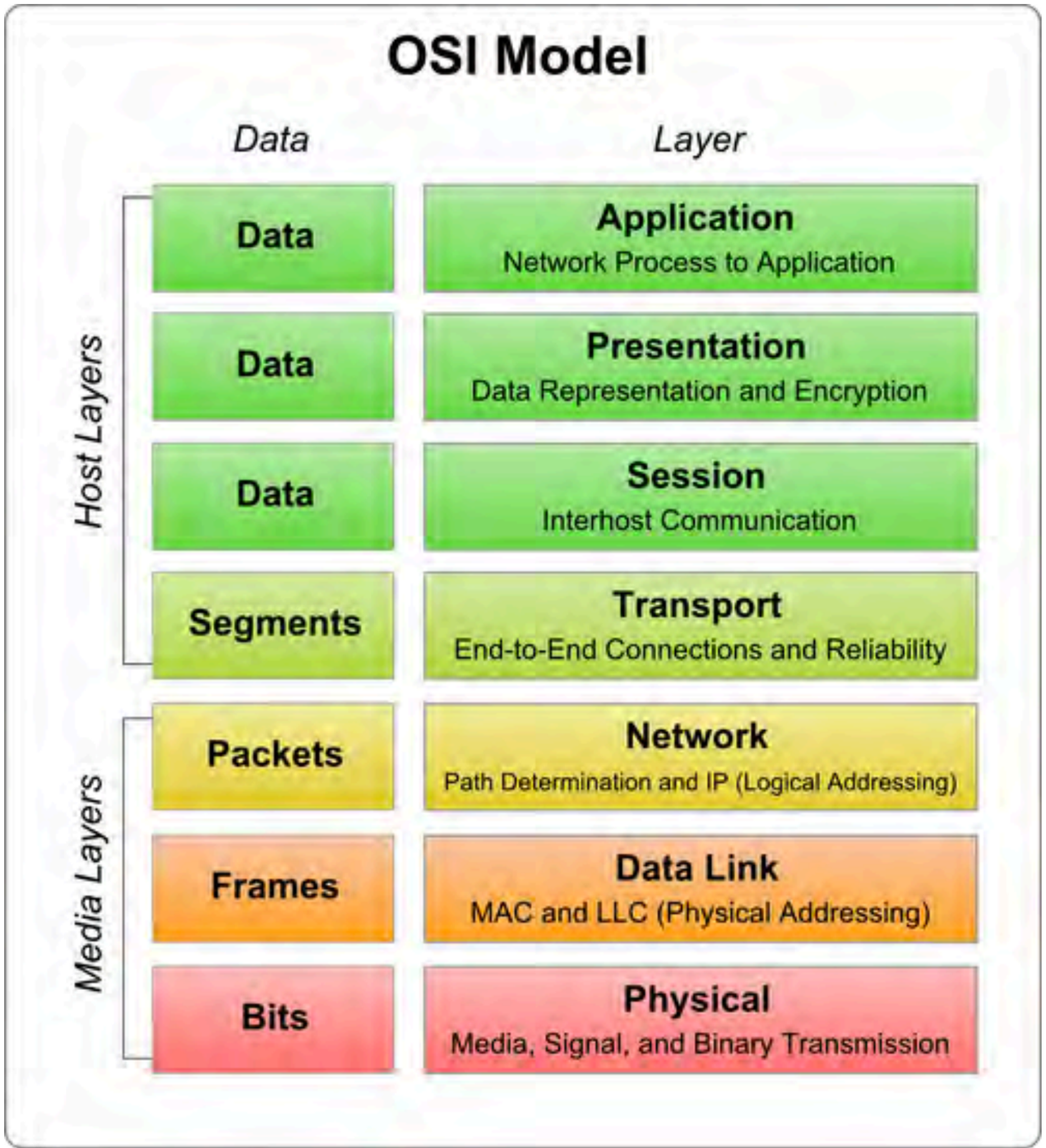
There are 2 major usage modes for `trackerjacker`: **map** mode and **track** mode:

Map mode example

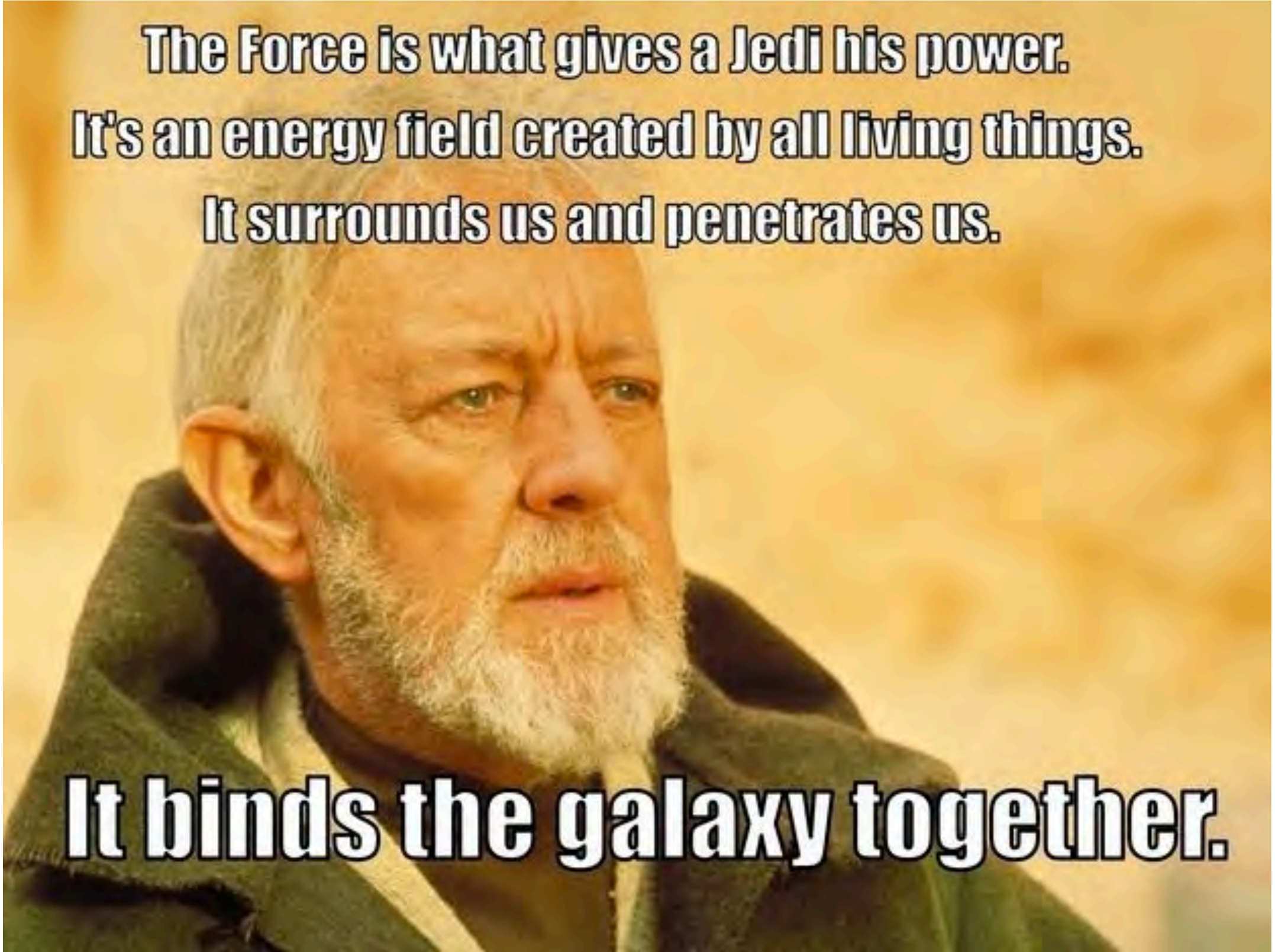
Map mode is used to find the Access Points and Devices within the range. Think of it like `nmap` for raw 802.11 mode.

HackRF One
GREAT SCOTT GADGETS
ALFA
www.alfa.com.tw
Columbia

We'll be exploring the Physical layer



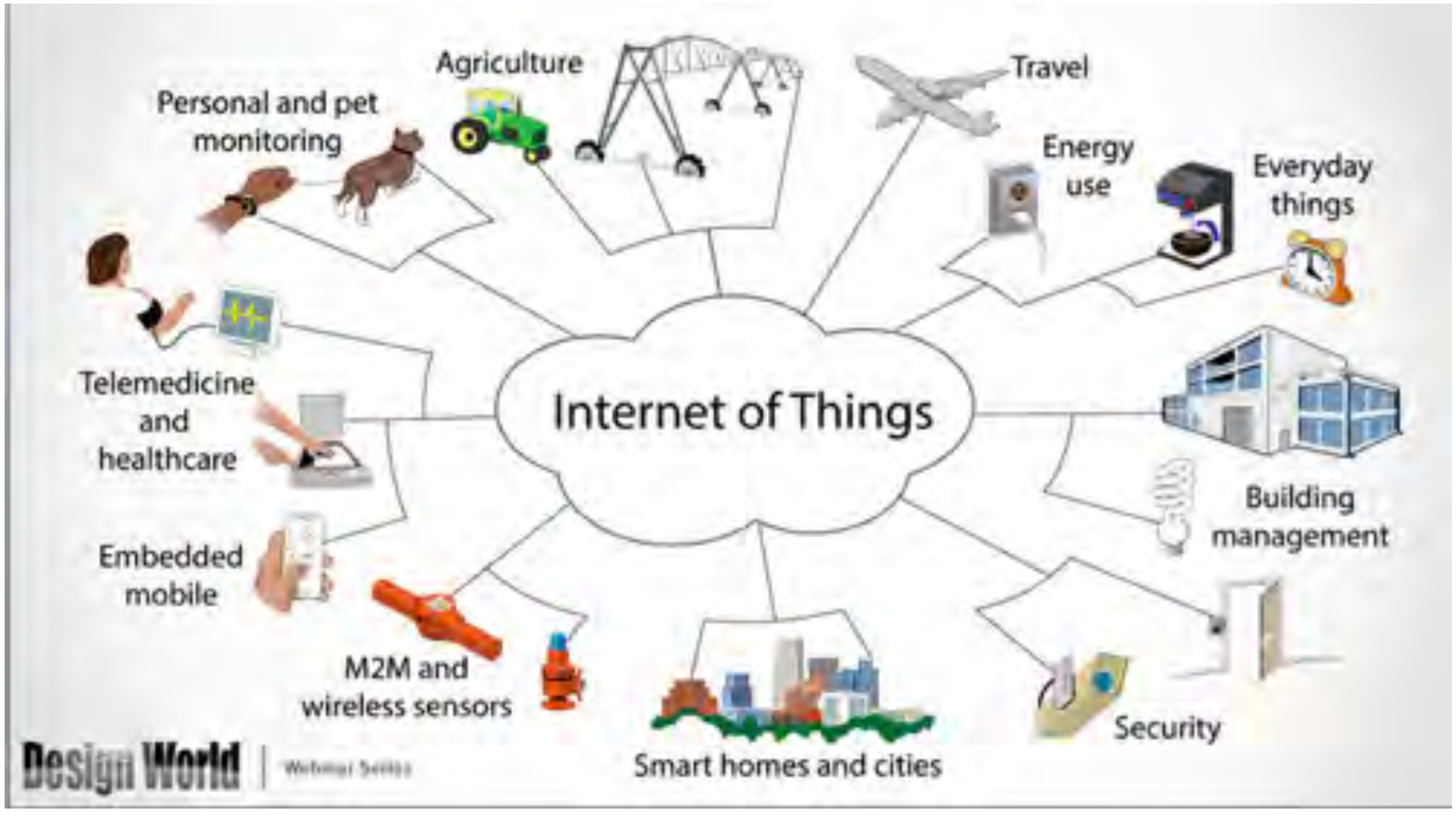
Harness the invisible energy all around us!


The Force is what gives a Jedi his power.
It's an energy field created by all living things.
It surrounds us and penetrates us.
It binds the galaxy together.


2

Sorceress Skills

IoT: +5000% attack damage to all wireless hacking skills



Things that work through radio:

- "Radio" (AM, FM)
- TV
- Cell phones
- Wifi
- Bluetooth
- GPS
- Wireless security systems
- Any form of Wireless IoT device
- SCADA systems / large industrial equipment

Milwaukee Metropolitan Sewerage District

Frequency	License	Type	Tone	Alpha Tag	Description	Mode	Tag
45.40000	KLR280	RM	173.8 PL	MMSD	MMSD Lowband repeater	FM	Public Works
453.45000	KEY991	RM	131.8 PL	MMSD	MMSD	FM	Public Works
453.61250		RM		MMSD	MMSD	FMN	Public Works
453.62500	KEY991	RM	131.8 PL	MMSD	MMSD	FM	Public Works
453.80000	KEY991	RM	311 DPL	MMSD	MMSD	FM	Public Works
453.47500	WPZP631	F		MMSD Data	SCADA Data	Telm	Data
453.75000	WPZP657	F		MMSD Data	SCADA Data	Telm	Data
453.85000	WPZP631	F		MMSD Data	SCADA Data	Telm	Data
453.92500	WPZP631	F		MMSD Data	SCADA Data	Telm	Data

Software-Defined Radio (SDR)

HackRF One

LimeSDR

RTL-SDR

In the Wizard/Sorceress metaphor, this is your wand (along with the antenna, of course).

- Sidenote: when using radios, your body can actually become a part of the antenna, which is very like magical lore - THE ENERGY IS FLOWING THROUGH YOU!

Demos

Unlocking Car (video)

https://www.youtube.com/watch?v=Q-OlgVLHIDs

Jamming (live)

./jam_narrow.py 440440000

Controlling outlet (live)

python3 ask_modulate_radio_signal.py -c raw_data/outlet_c2_on.json -o /tmp/radio_signal.pcm

./transmit_signal.py -f /tmp/radio_signal.pcm -t 315000000

Audacity File Edit View Transport Tracks Generate Effect Analyze

New ⌘N
Open... ⌘O
Open Recent ▶

Close ⌘W
Save Project ⌘S
Save Project As...
Save Compressed Copy of Project...
Check Dependencies...

Edit Metadata...

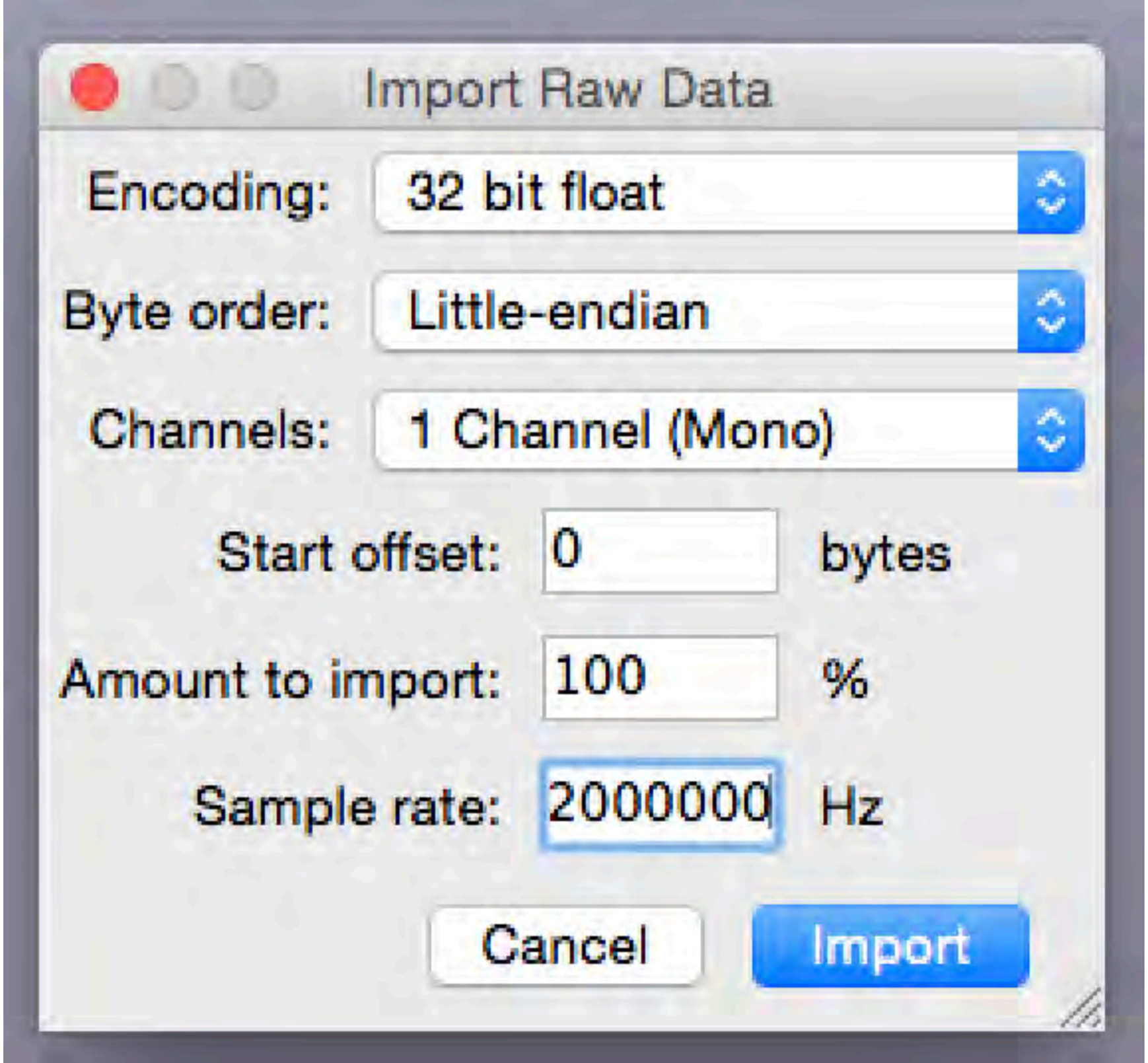
Import ▶
Audio... ⇧⌘I
Labels...
MIDI...
Raw Data...

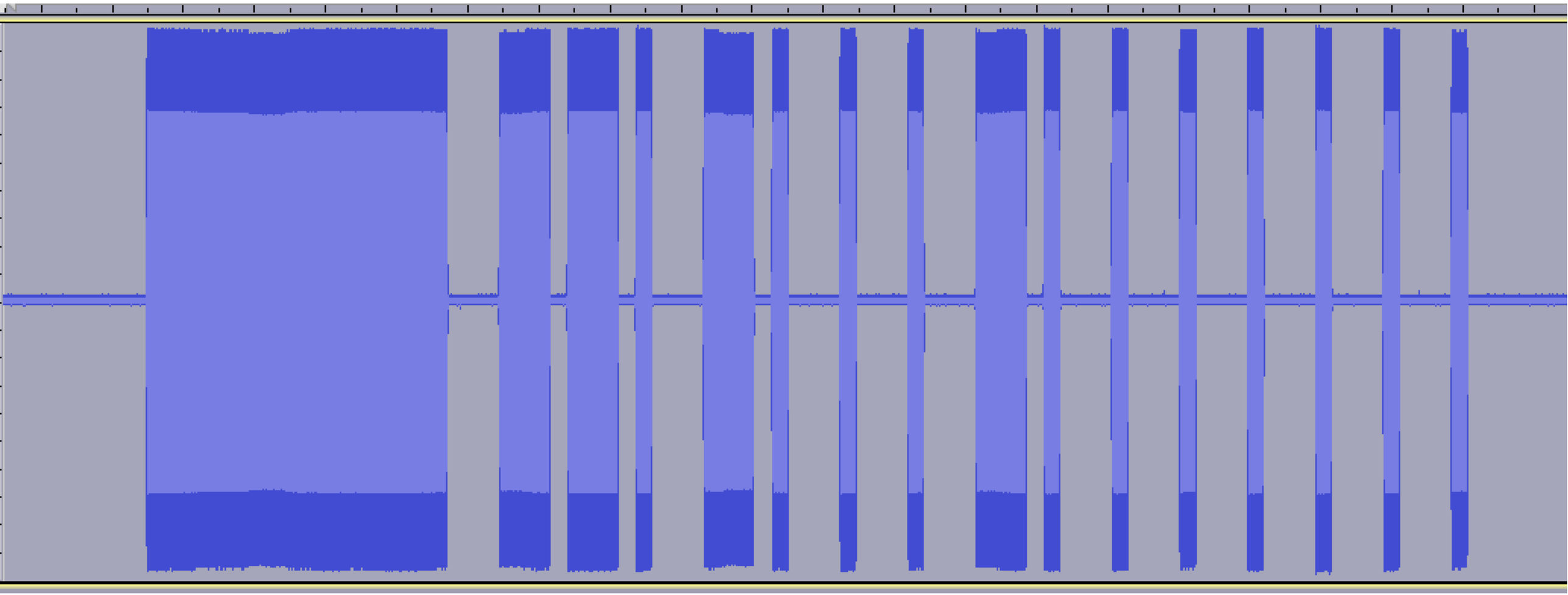
Export... ⇧⌘E
Export Selection...

Export Labels...
Export Multiple... ⇧⌘L
Export MIDI...
Apply Chain...
Edit Chains...

Page Setup...
Print...

L
R
-36 -24 -12 0
Built-in Microph
- 1.0
3.0
4.0


Import Raw Data
Encoding:
32 bit float
Byte order:
Little-endian
Channels:
1 Channel (Mono)
Start offset:
0
bytes
Amount to import:
100
%
Sample rate:
2000000
Hz
Cancel
Import



Save As: jeep_unlock

Tags:

gnu_radio

Search

Previous 30 Days

January

gnu_radio

2015

metro_coc

play_fort

play_fort~

WebEx

2014

design

DevelopmentorLabs.zip

Health

Spiderlogic

Virtual Machines

2013

Microsoft User Data

RDC Connections

to_organize

Today

record_signal.grc

Previous 7 Days

visualize_file.grc

January

audio_frequ…enera

captures

hackrf_record_315M

hackrf_repla…lock_

noise_and_filters.gr

radio_record.grc

radio_replay.grc

record_replay.grc

repos

signal_generator.grc

signal_jammer_narr

signal_jammer_no_f

signal_jammer.grc

top_block.py

Hive

Cloud Drive

ropbox

pplications

esktop

ocuments

ownloads

oogle Drive

piderOak Hive

emote Disc

omecontroller

lyBookLive

Forma

✓ Other uncompressed files

AIFF (Apple) signed 16 bit PCM

WAV (Microsoft) signed 16 bit PCM

GSM 6.10 WAV (mobile)

Options...

ew Folder

Cancel

Hive

play_fort

January

audio_frequ…eneration.grc

Specify Uncompressed Options

Uncompressed Export Setup

Header: RAW (header-less)

Encoding: 32 bit float

(Not all combinations of headers and encodings are possible.)

Cancel OK

cord_315MHz.grc

pla…lock_jeep.grc

_filters.grc

ord.grc

lay.grc

play.grc

nerator.grc

nmer_narrow.grc

nmer_no_filter.grc

signal_jammer.grc

top_block.py

RDC Connections

to_organize

Format: Other uncompressed files

Options...

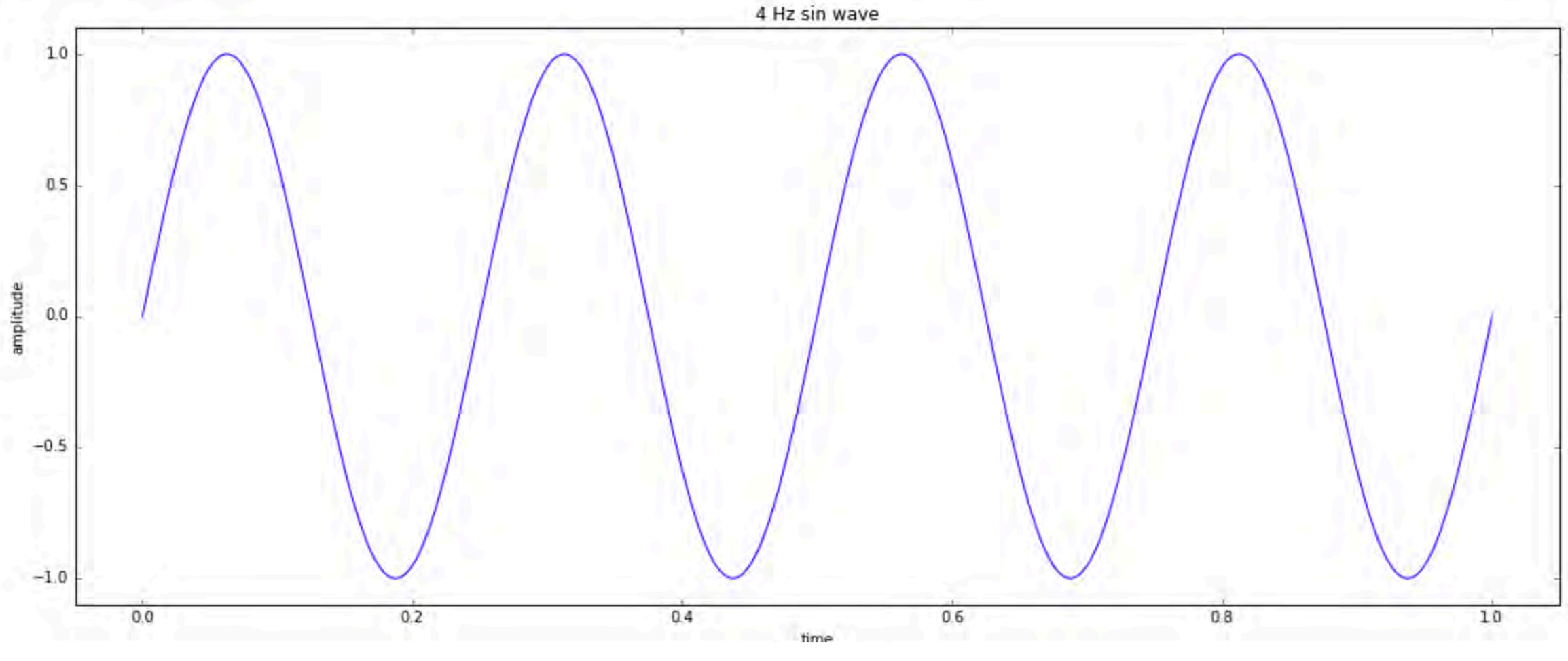
Cancel Save

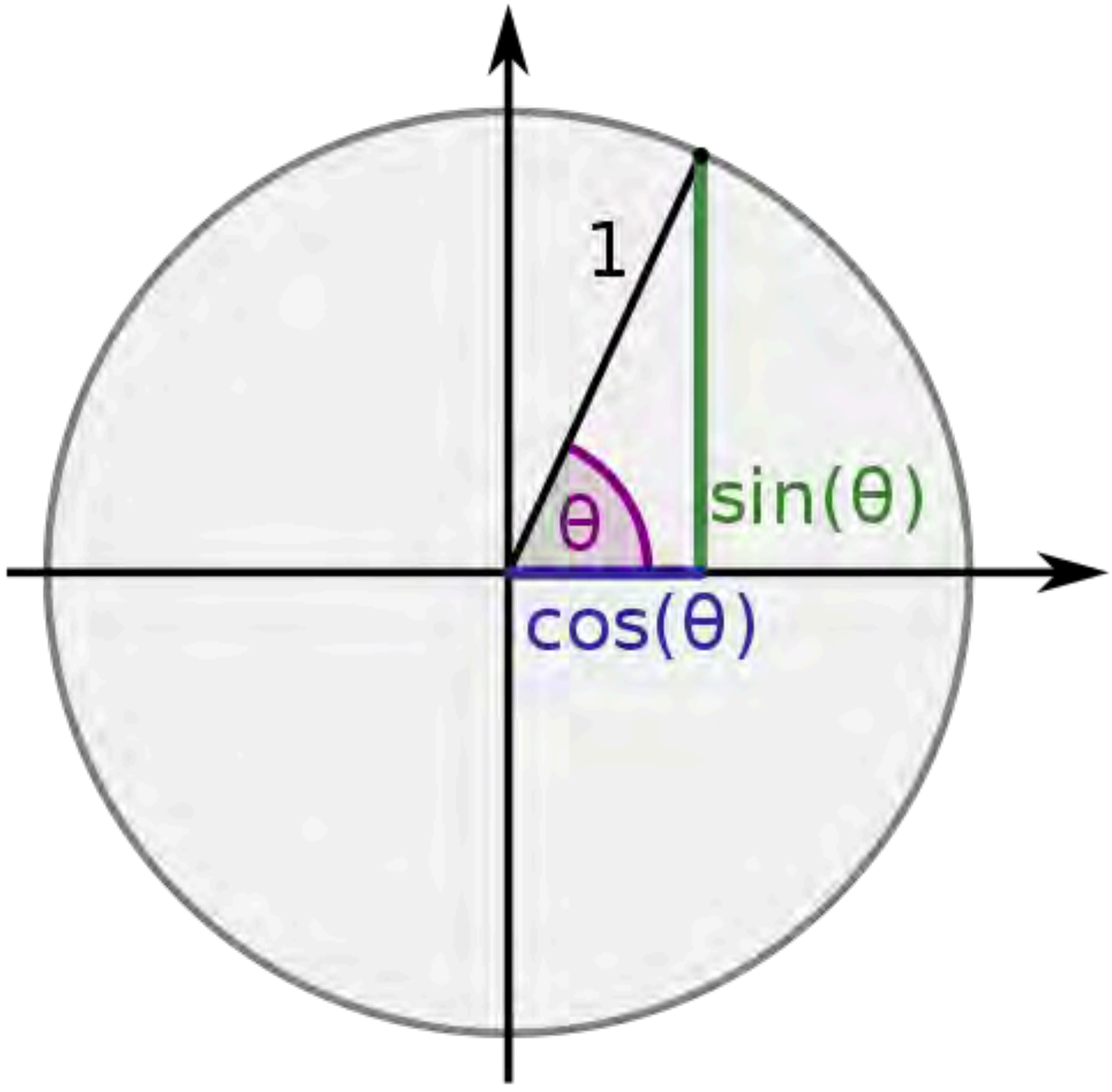
Understanding waves

What are waves?

In [21]:
```
samp_rate = 1000
len_in_sec = 1
t = np.linspace(0, len_in_sec, samp_rate * len_in_sec)
hz_4 = 1*np.sin(4 * 2 * np.pi * t)

setup_graph(title='4 Hz sin wave', x_label='time', y_label='amplitude', fig_size=(18,7))
plt.plot(t, hz_4)
plt.margins(0.05)
```



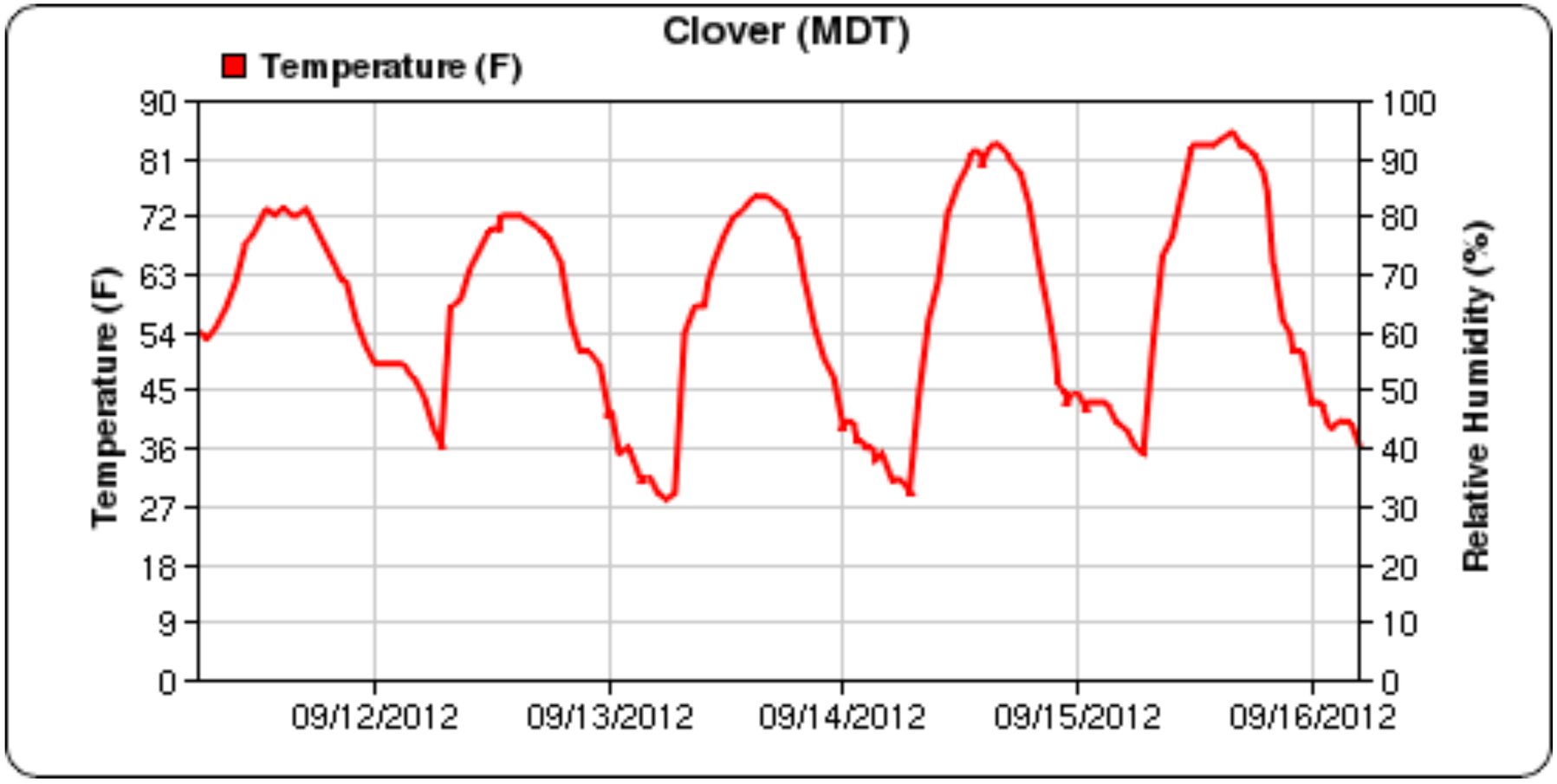

1
θ
sin(θ)
cos(θ)

Interesting things about waves

- They are everywhere
- Epitome of change
- Superposition principle (wave convolution)
- Convoluted waves can be deconvoluted
- Orthogonality of waves of differing frequencies
- Relation to e

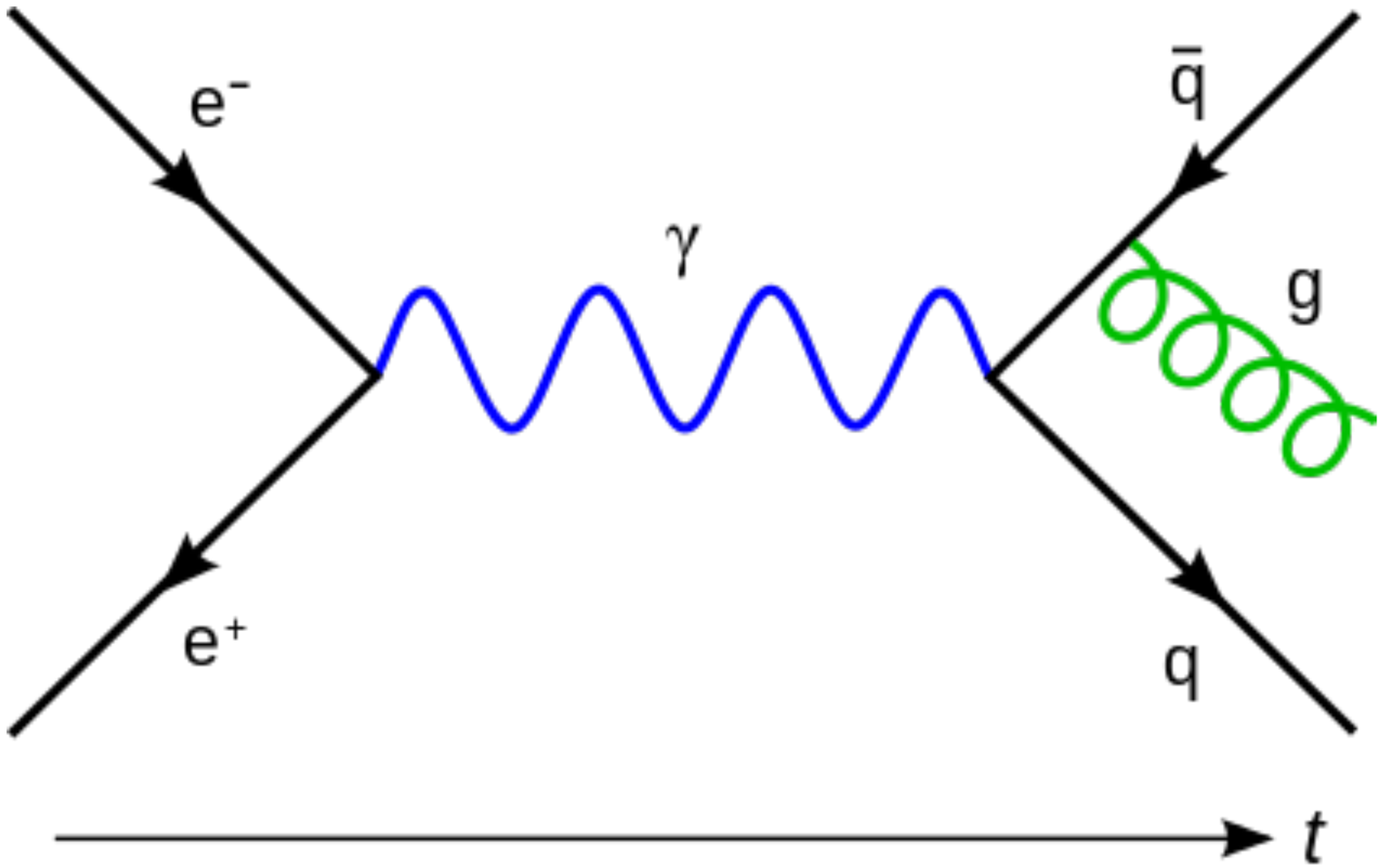
Waves are found everywhere!

- Anywhere there is circular motion or vibration, there are waves.
- All wireless communication
    - Sounds waves
    - Radio waves
    - Infrared, microwave, etc
- Motion of pendulums and springs
- Light from the sun at a given place on earth
- Temperature on earth
- Patterns of the tides


Clover (MDT)
Temperature (F)
90
81
72
63
54
45
36
27
18
9
0
Temperature (F)
100
90
80
70
60
50
40
30
20
10
0
Relative Humidity (%)
09/12/2012
09/13/2012
09/14/2012
09/15/2012
09/16/2012

All over moden physics

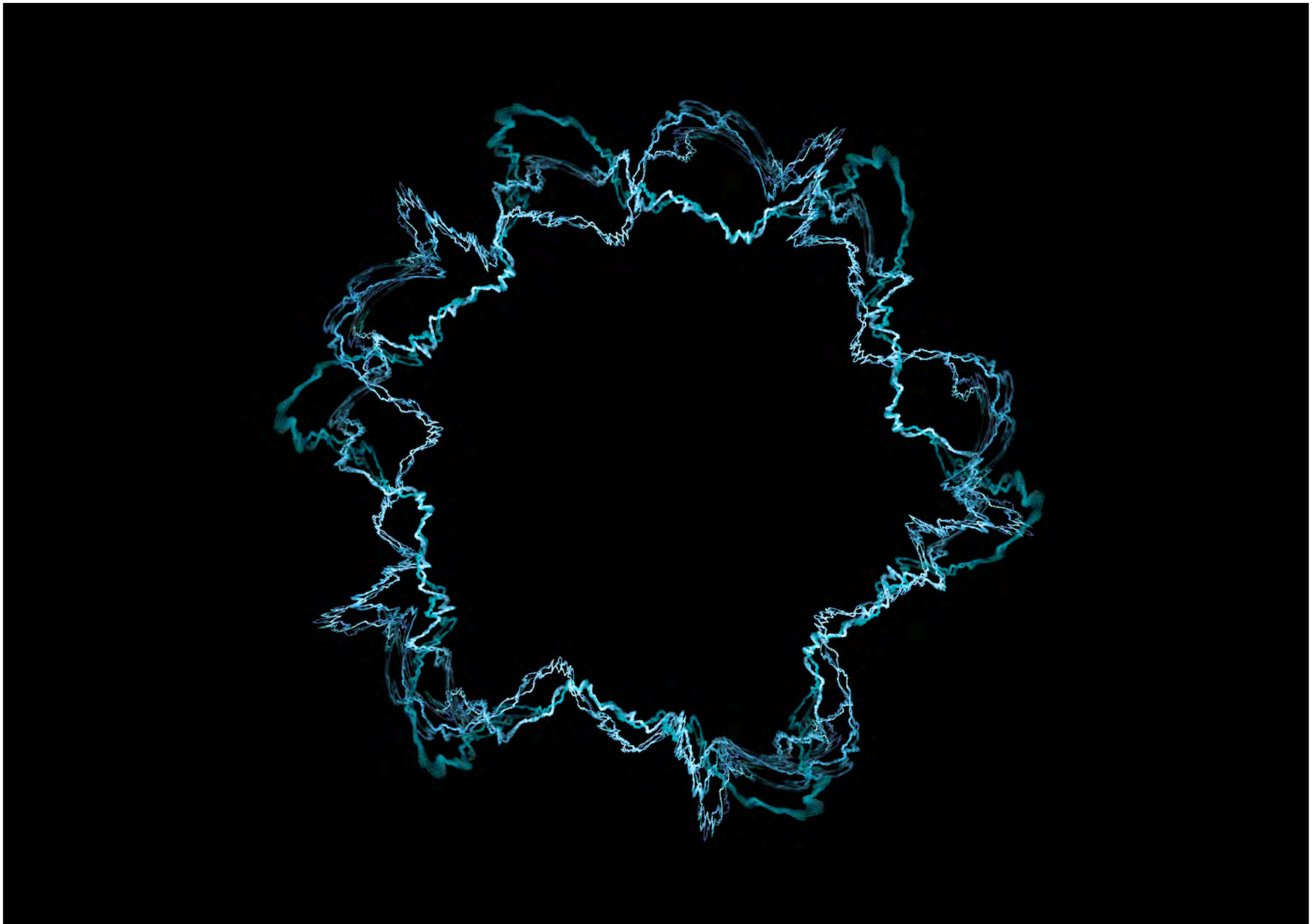
Quantum Field Theory posits that particles are excitations (waves) in various fields.



We know from Einstein that matter and energy are related, and since waves are the epitome of energy, this makes sense.

Possibly the basis for all matter

String Theory posits that everything is made out of small VIBRATING strings

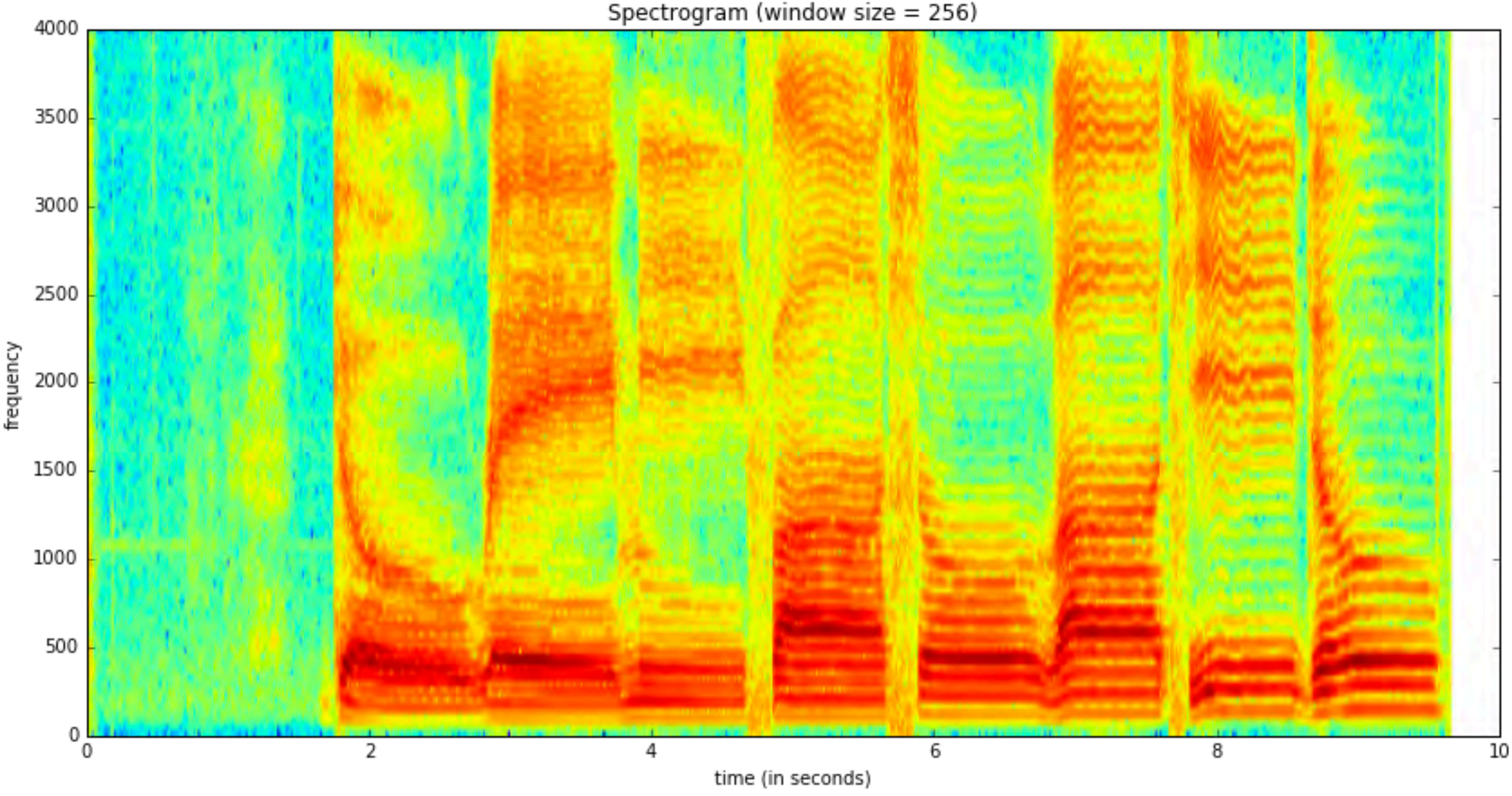


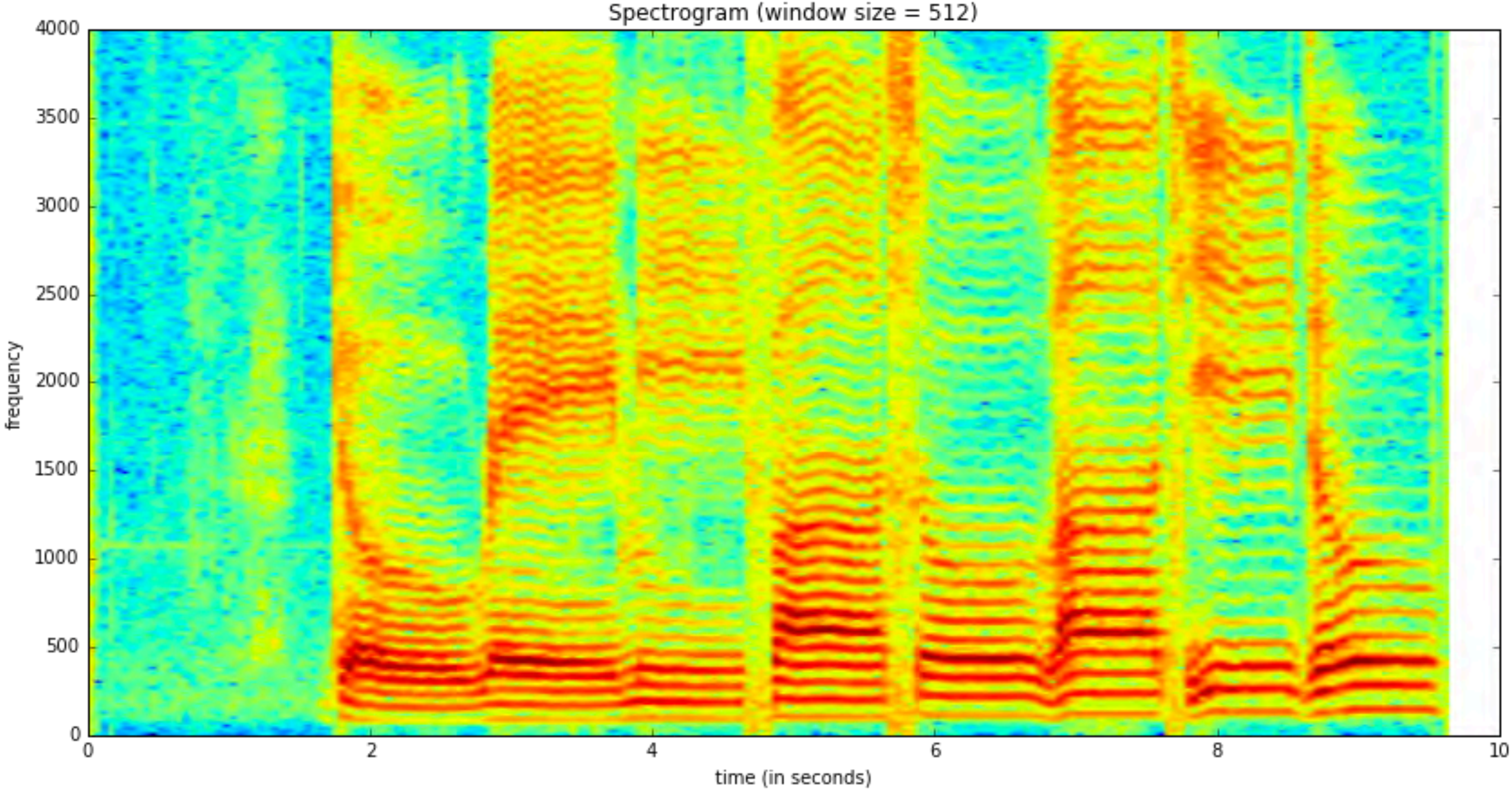
Uncertainty Principle

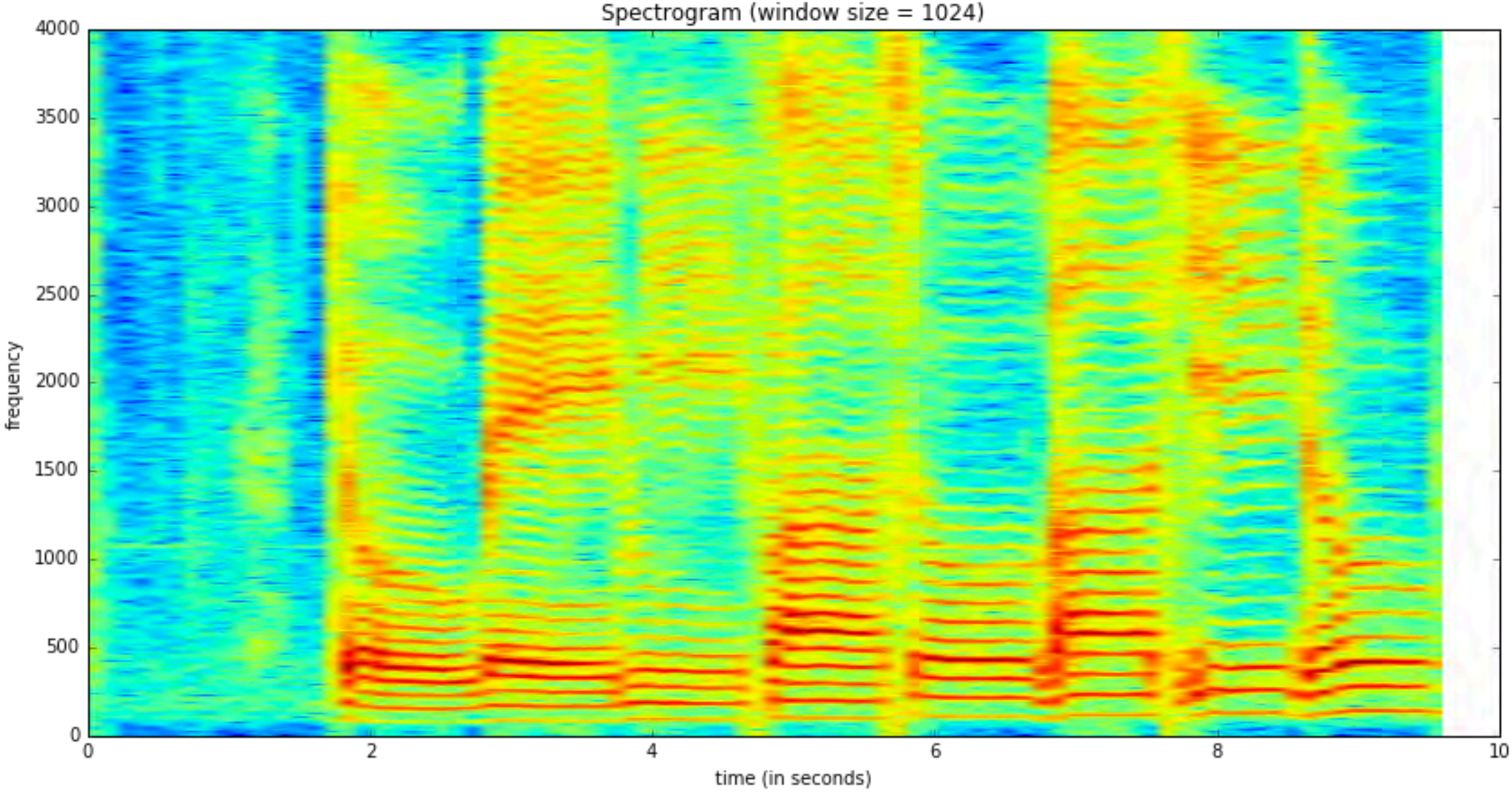
I accidentally stumbled on the Uncertainty Principle when doing audio analysis

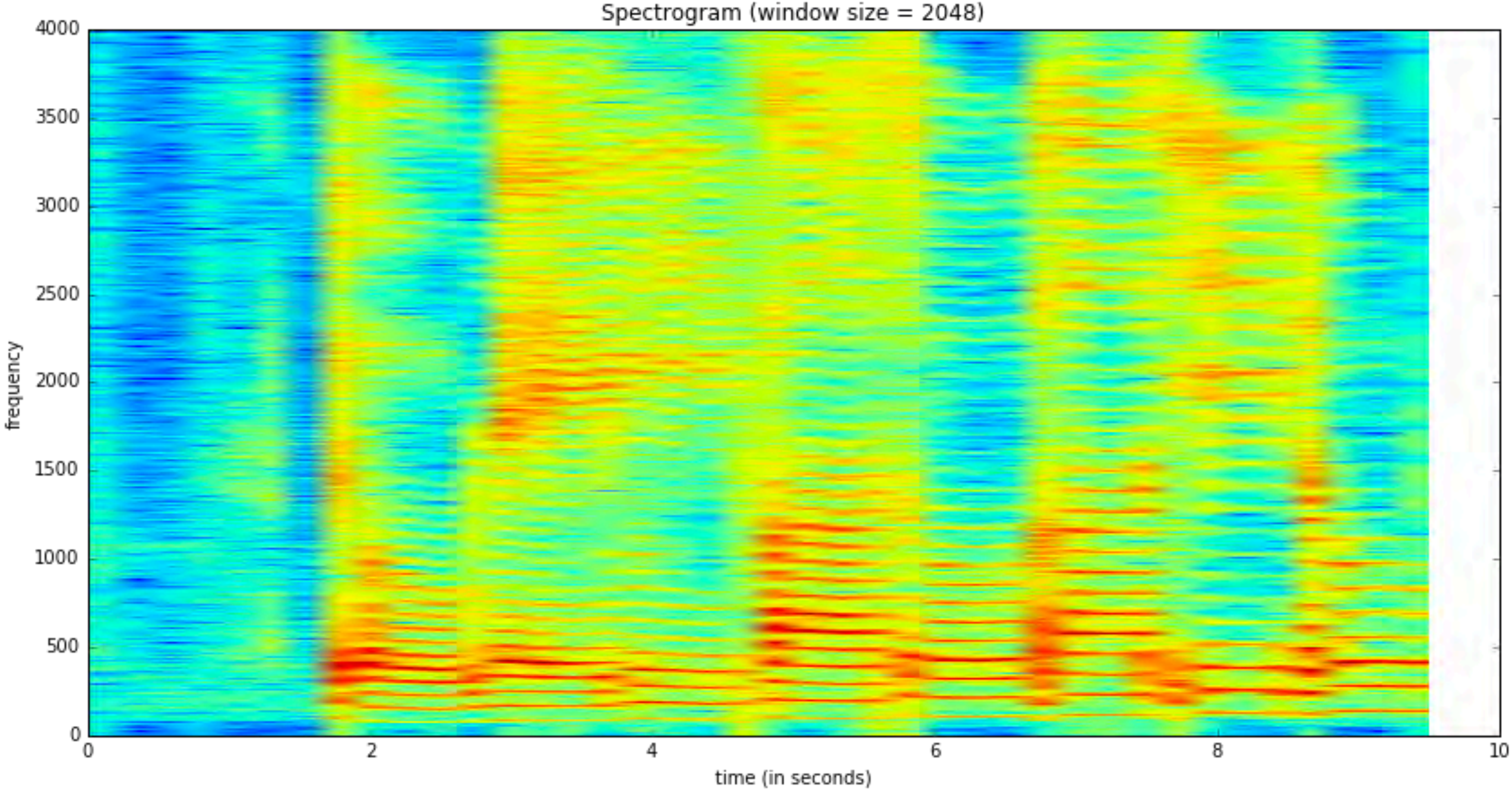
The more samples you take, the more certain you can be of the frequencies contained

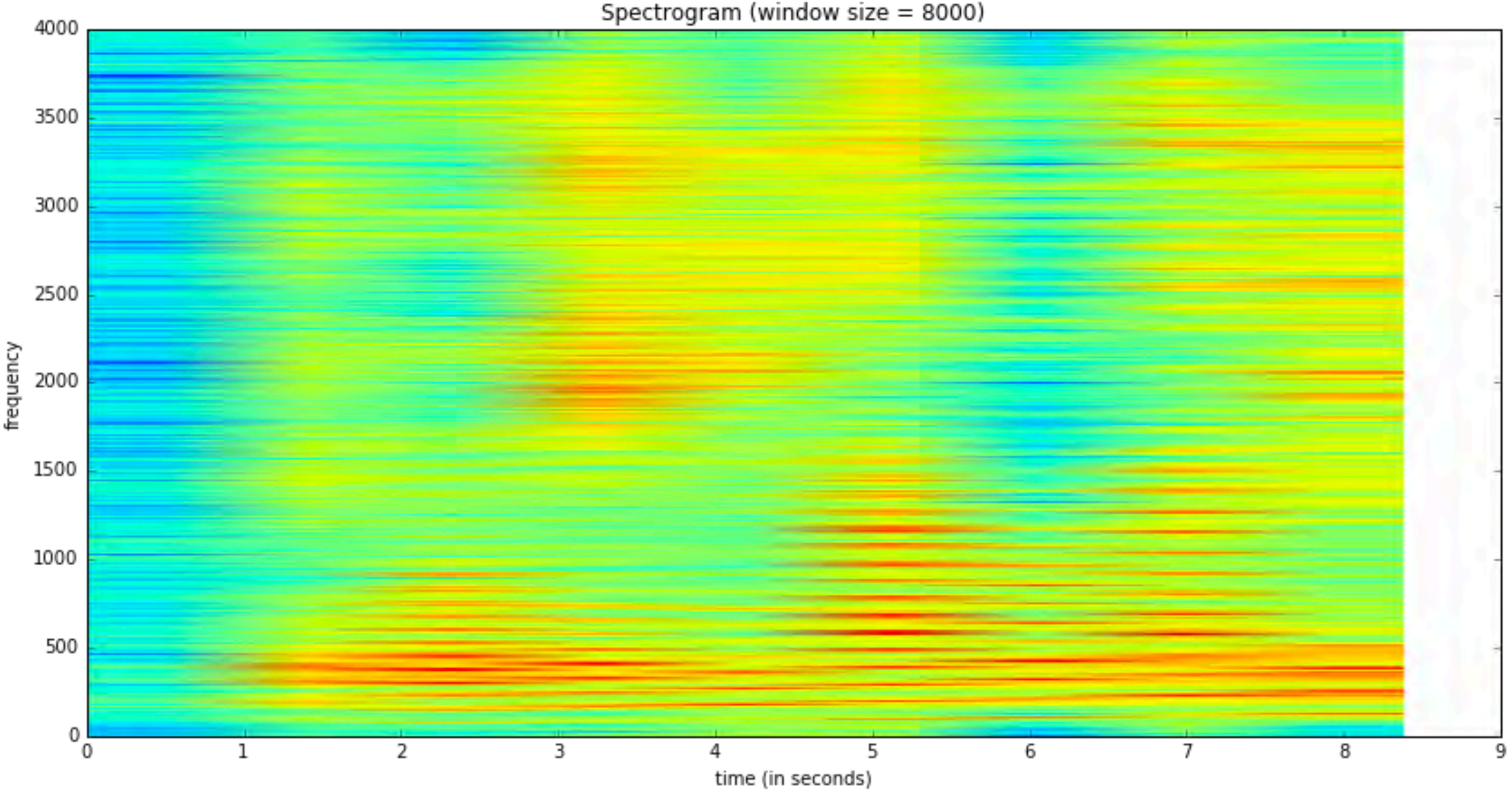
But the more samples you take, the less localized in time you are.


Spectrogram (window size = 256)
frequency
4000
3500
3000
2500
2000
1500
1000
500
0
0
2
4
6
8
10
time (in seconds)


Spectrogram (window size = 512)
frequency
4000
3500
3000
2500
2000
1500
1000
500
0
0
2
4
6
8
10
time (in seconds)


Spectrogram (window size = 1024)
frequency
4000
3500
3000
2500
2000
1500
1000
500
0
0
2
4
6
8
10
time (in seconds)


Spectrogram (window size = 2048)
frequency
4000
3500
3000
2500
2000
1500
1000
500
0
0
2
4
6
8
10
time (in seconds)


Spectrogram (window size = 8000)
frequency
4000
3500
3000
2500
2000
1500
1000
500
0
0
1
2
3
4
5
6
7
8
9
time (in seconds)

Application: ***So understanding waves helps you understand the universe!***

Waves are the epitome of change

Or to put it another way, "energy".

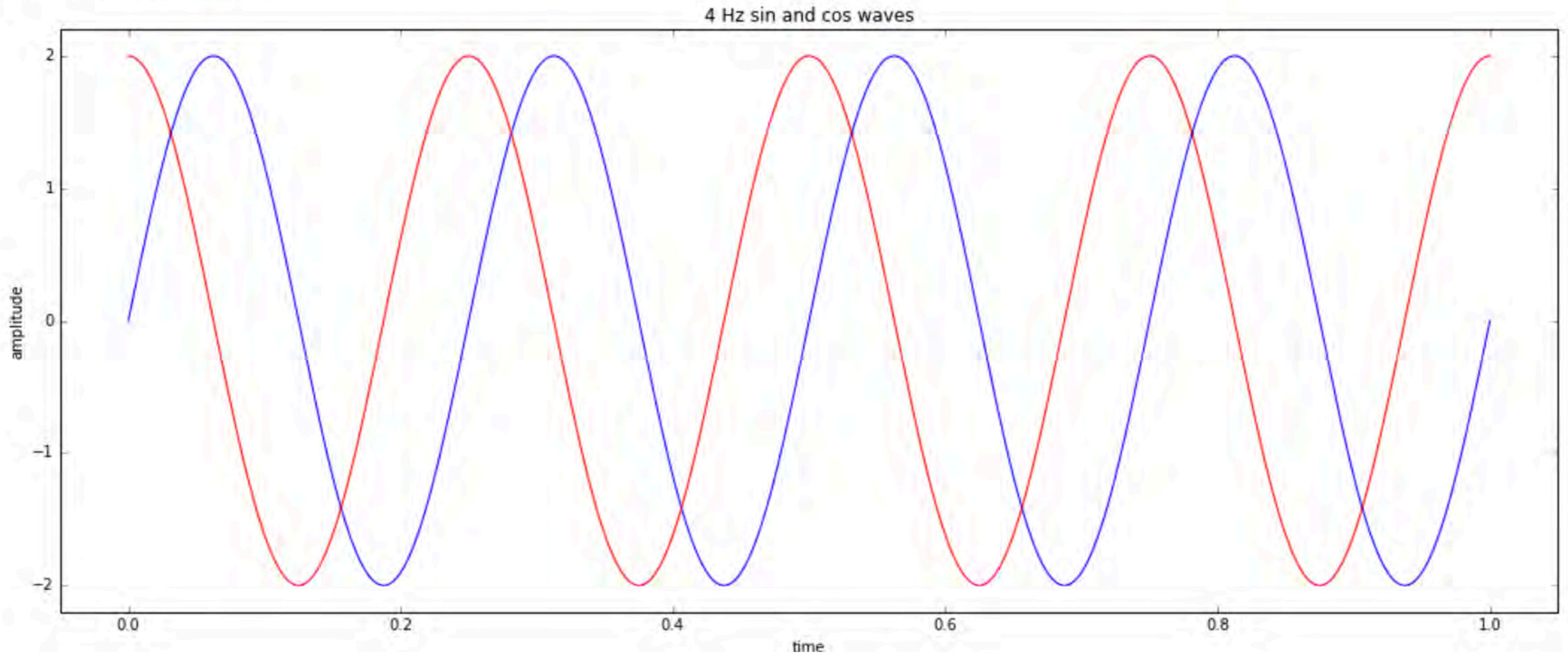
Calculus deals with change through the "derivative" - the rate of change of a function.

$$\frac{d}{dt}sin(x) = cos(x)$$

$$\frac{d}{dt}cos(x) = -sin(x)$$

In [22]:
```
samp_rate = 1000
len_in_sec = 1
t = np.linspace(0, len_in_sec, samp_rate * len_in_sec)
sin_4hz = 2*np.sin(4 * 2 * np.pi * t)
cos_4hz = 2*np.cos(4 * 2 * np.pi * t)

setup_graph(title='4 Hz sin and cos waves', x_label='time', y_label='amplitude', fig_size=(18,7))
plt.plot(t, sin_4hz)
plt.plot(t, cos_4hz, color='red')
plt.margins(0.05)
```



Waves are one of the only graphs for which the derivative of the function is a version of itself!

The other one I know of is e^x, but there's a reason for that... waves and e are intimately related in Euler's Formula, e^ix = cos(x) + i sin(x).

More on that in a minute.

Superposition principle

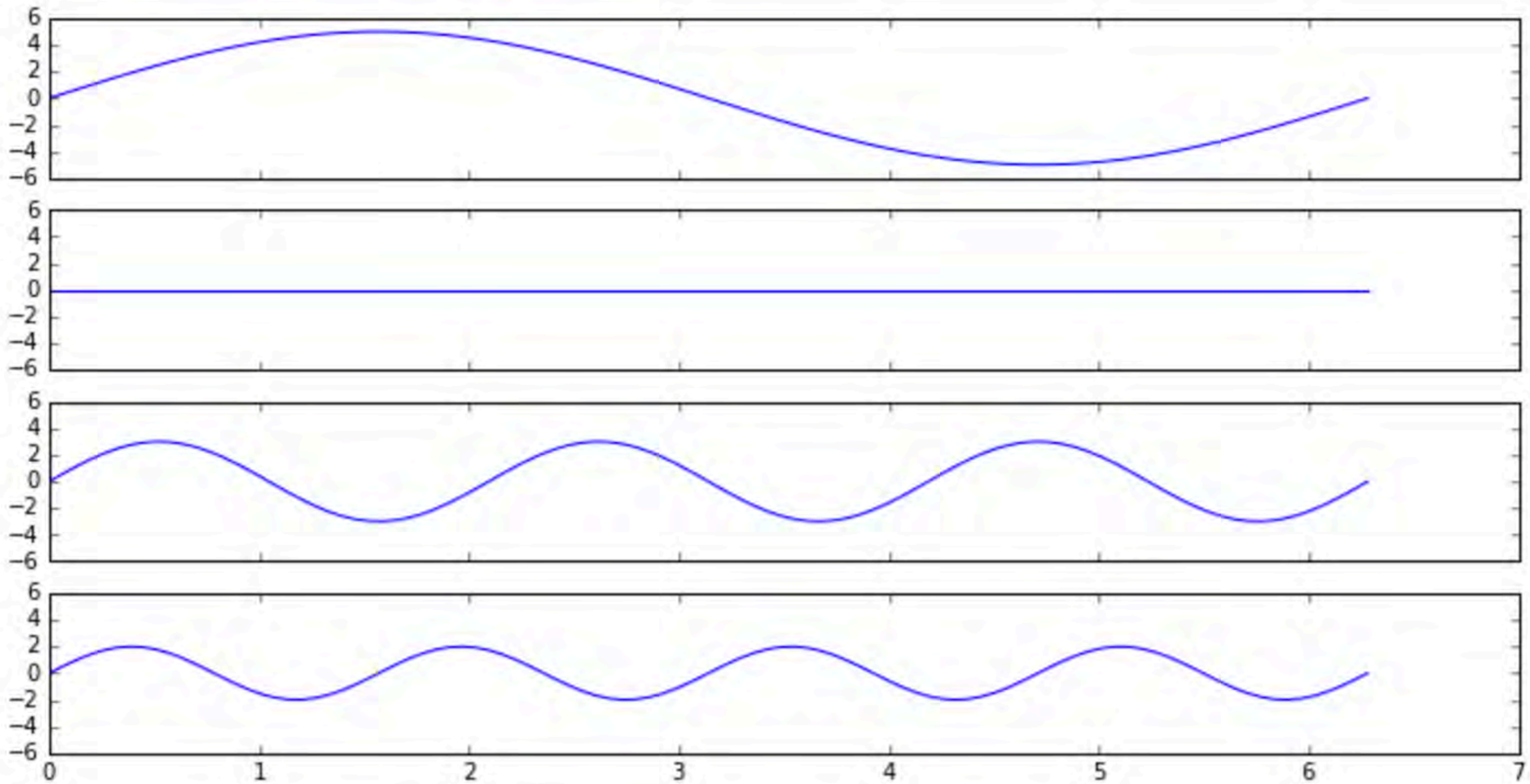
How can you hear many things at once?

How can there be multiple radio waves in the air at the same time which can be tuned into?

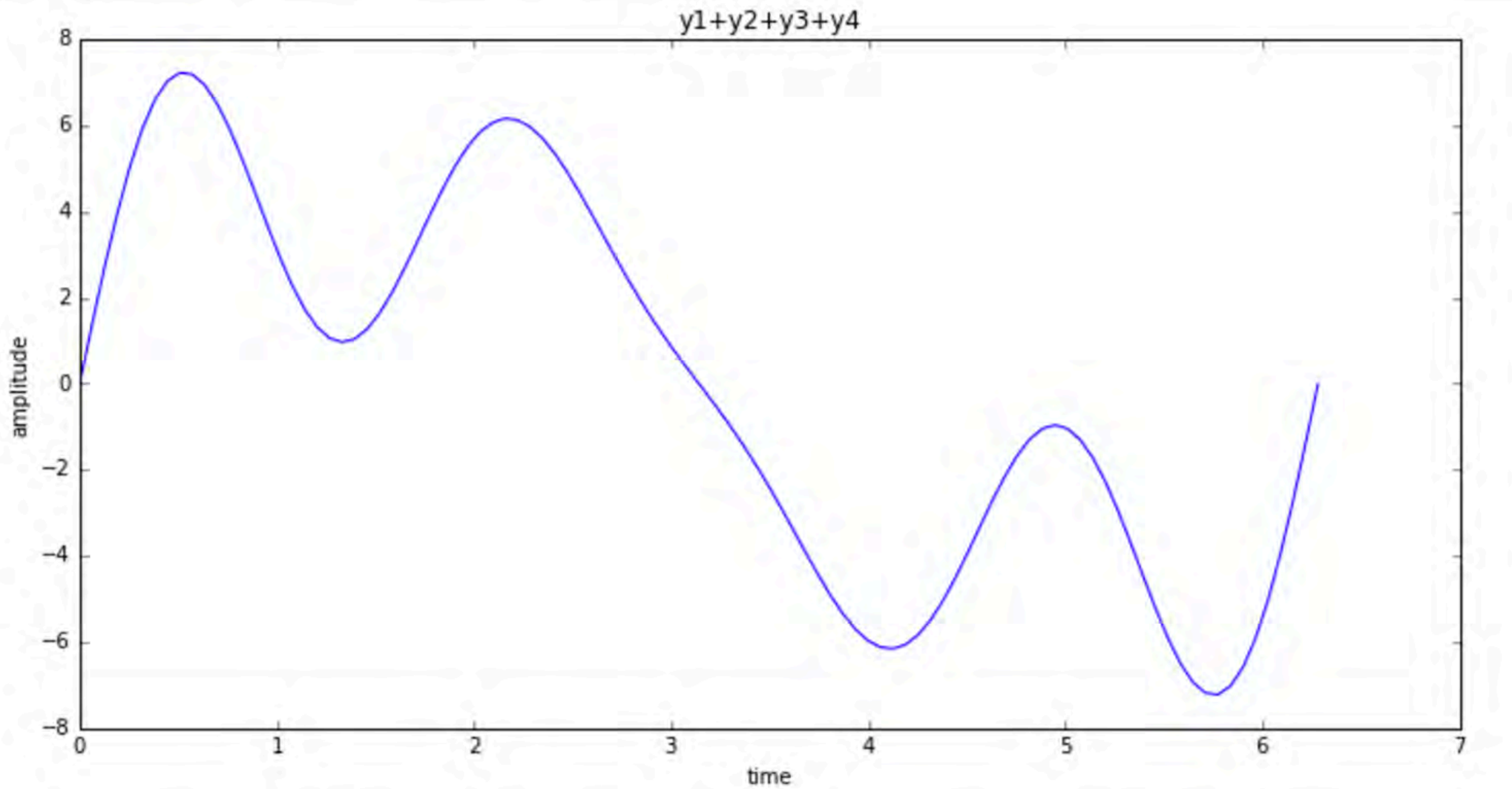
Superposition principle: waves are added together to form more complicated wave patterns.

In [27]:
```
# Two subplots, the axes array is 1-d
x = np.linspace(0, 2 * np.pi, 100)
y1 = 5 * np.sin(x)
y2 = 0 * np.sin(2*x)
y3 = 3 * np.sin(3*x)
y4 = 2 * np.sin(4*x)

f, axarr = plt.subplots(4, sharex=True, sharey=True)
f.set_size_inches(12,6)
axarr[0].plot(x, y1)
axarr[1].plot(x, y2)
axarr[2].plot(x, y3)
axarr[3].plot(x, y4)
_ = plt.show()
```



In [28]:
```
setup_graph(x_label='time', y_label='amplitude', title='y1+y2+y3+y4', fig_size=(12,6))
convoluted_wave = y1 + y2 + y3 + y4
_ = plt.plot(x, convoluted_wave)
```



Relation to radio hacking: The superposition principle is how there can be many different radio signals in the air at the same time.

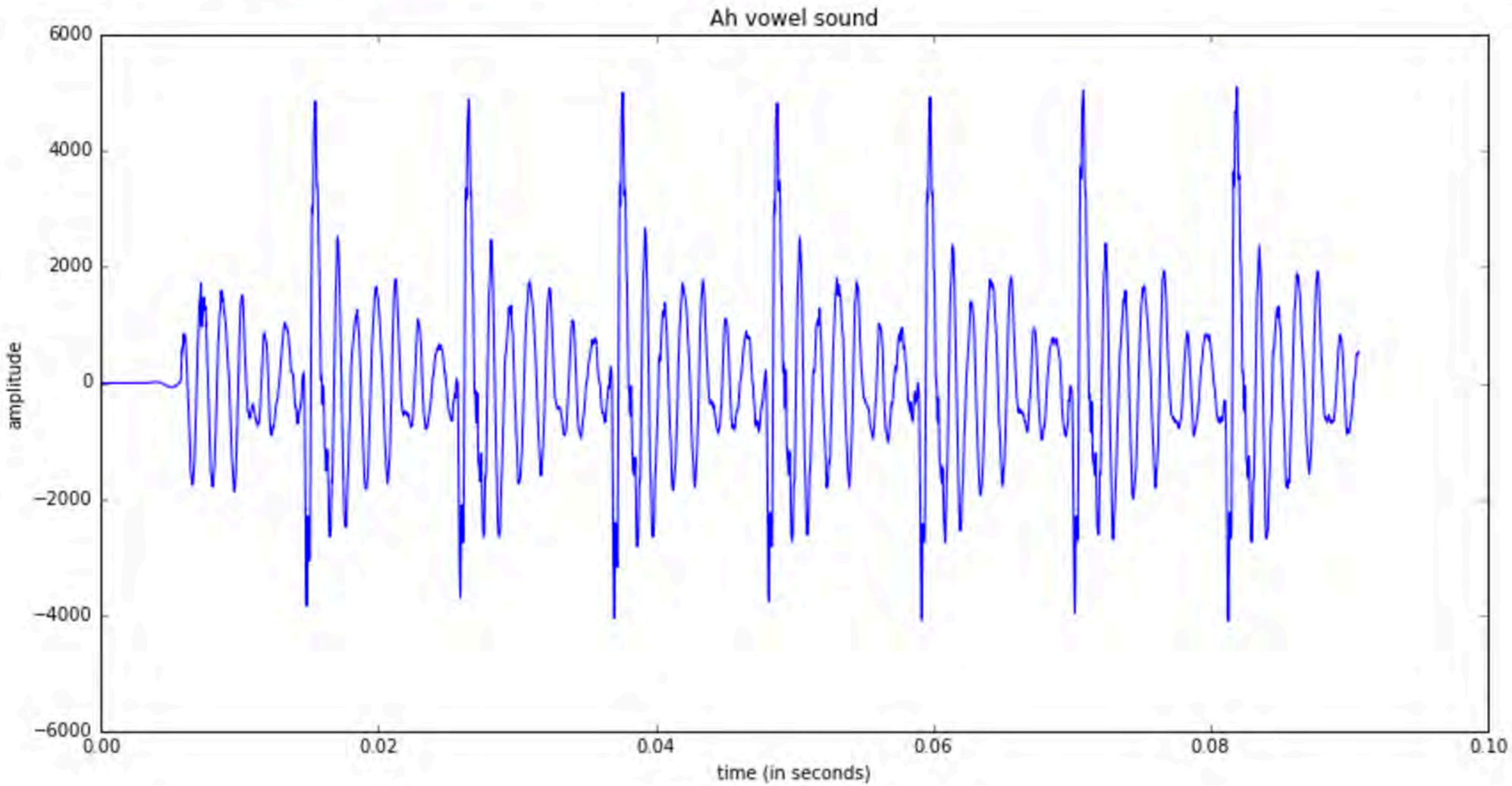
Deconvolution is possible

Because waves of differing frequencies are orthogonal, a convoluted wave (made of multiple frequencies) can be broken into each of its component frequencies.

This can be done with the Fourier Transform.

Time domain

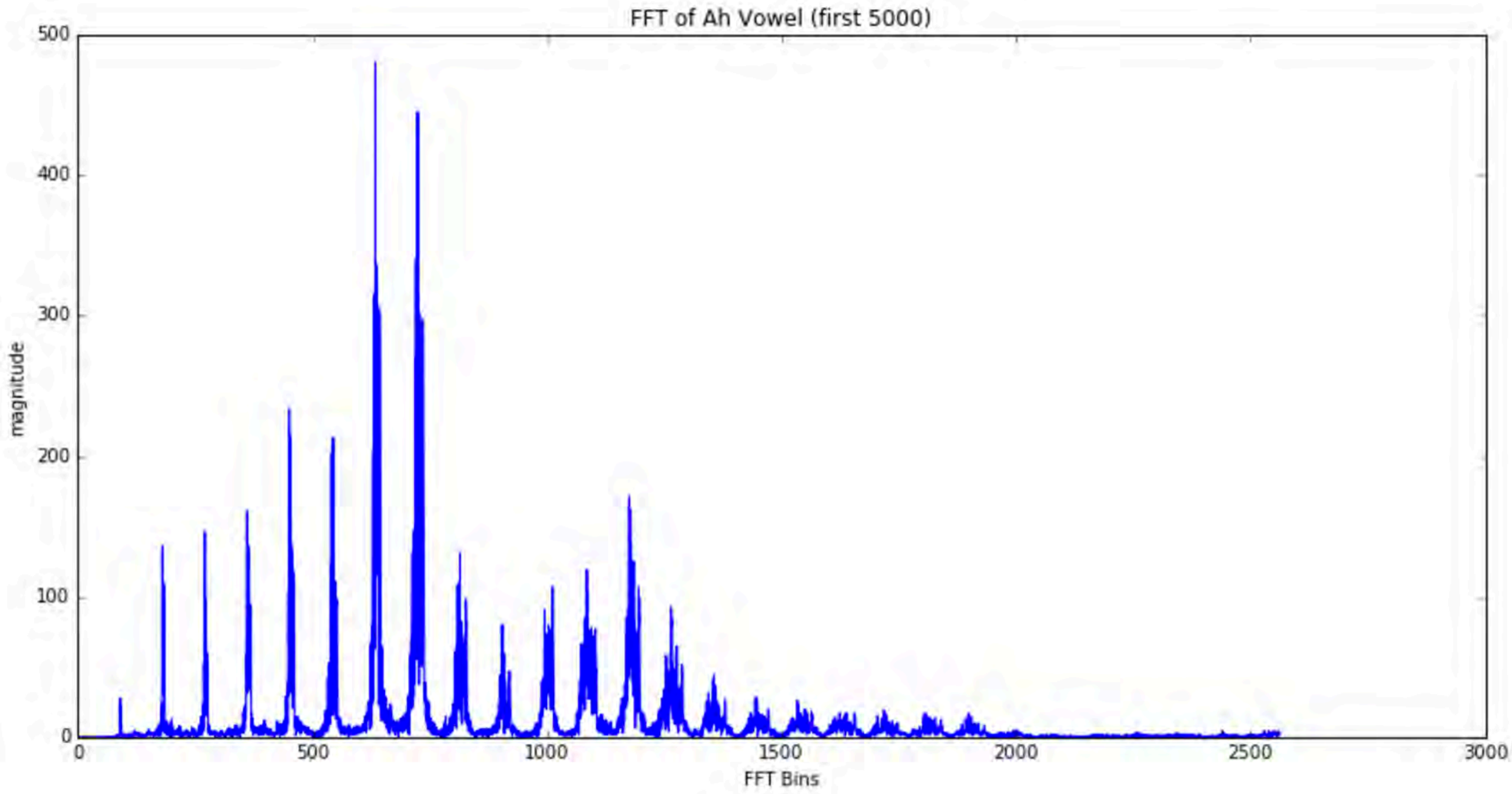
In [35]:
```
setup_graph(title='Ah vowel sound', x_label='time (in seconds)', y_label='amplitude', fig_size=(14,7))
_ = plt.plot(time_array[0:4000], input_signal[0:4000])
```



Frequency domain

In [37]:
```
fft_out = np.fft.rfft(input_signal)
fft_mag = [np.sqrt(i.real**2 + i.imag**2)/len(fft_out) for i in fft_out]
num_samples = len(input_signal)
rfreqs = [(i*1.0/num_samples)*sample_rate for i in range(num_samples//2+1)]

setup_graph(title='FFT of Ah Vowel (first 5000)', x_label='FFT Bins', y_label='magnitude', fig_size=(14,7))
_ = plt.plot(rfreqs[0:5000], fft_mag[0:5000])
```



Relation to radio hacking: Deconvolution is basically how we "tune into" a particular radio frequency.

But how is this possible?

Say I have 10 numbers, and I "convolute" (add) them together - you couldn't break them back into the original numbers. So how can you do this with waves?

Orthogonality of waves of differing frequencies

Waves of differing frequencies are orthogonal to one another (dot product is zero):

In [30]:
```
t = np.linspace(0, 1, samp_rate * len_in_sec)
sin_8hz = 1*np.sin(8 * 2 * np.pi * t)
sin_12hz = 1*np.sin(12 * 2 * np.pi * t)
np.dot(sin_8hz, sin_12hz)
```

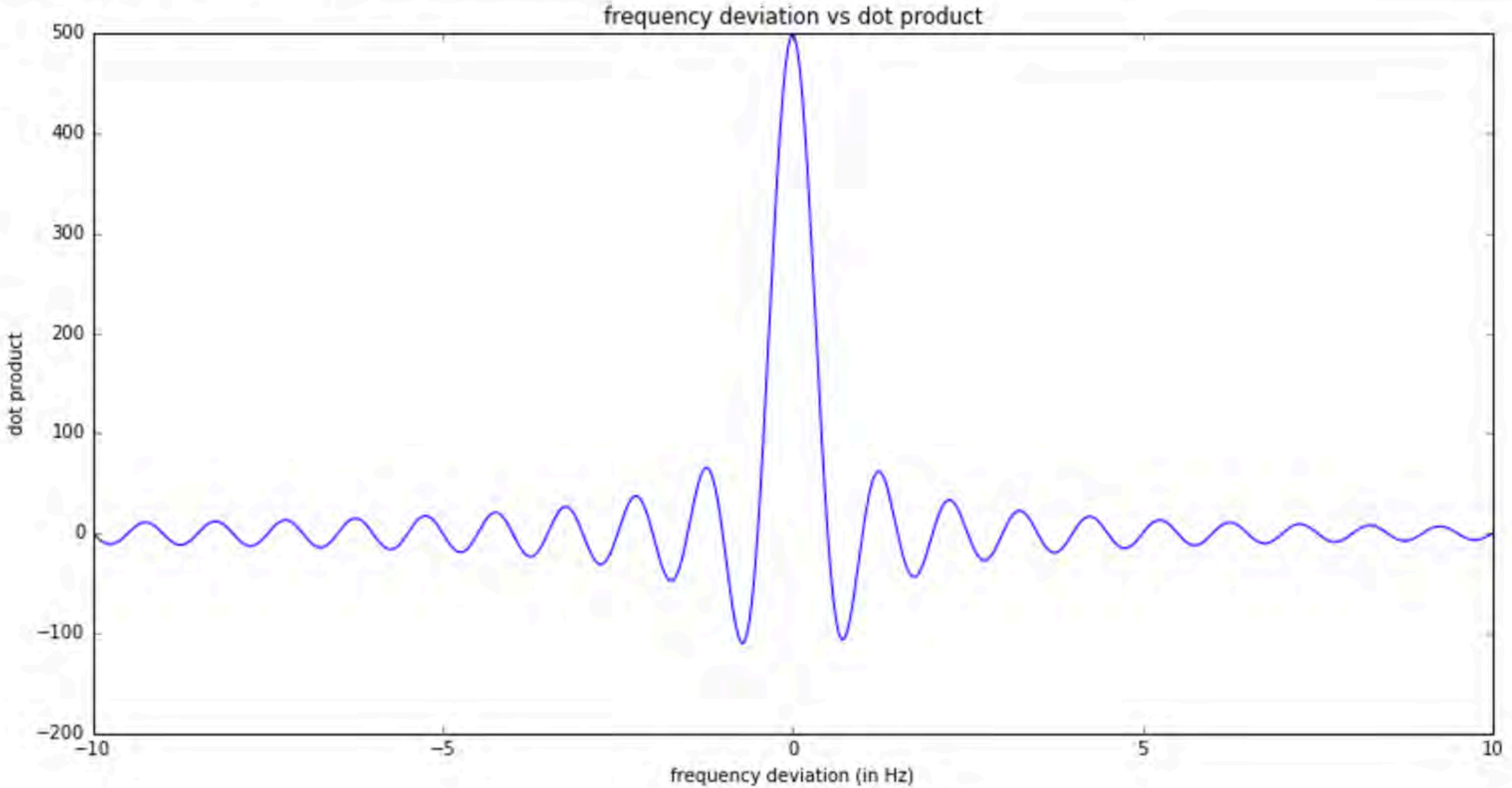
Out[30]:
```
2.0790660859582033e-15
```

Whereas, waves of the same (or close) frequencies hare larger dot products:

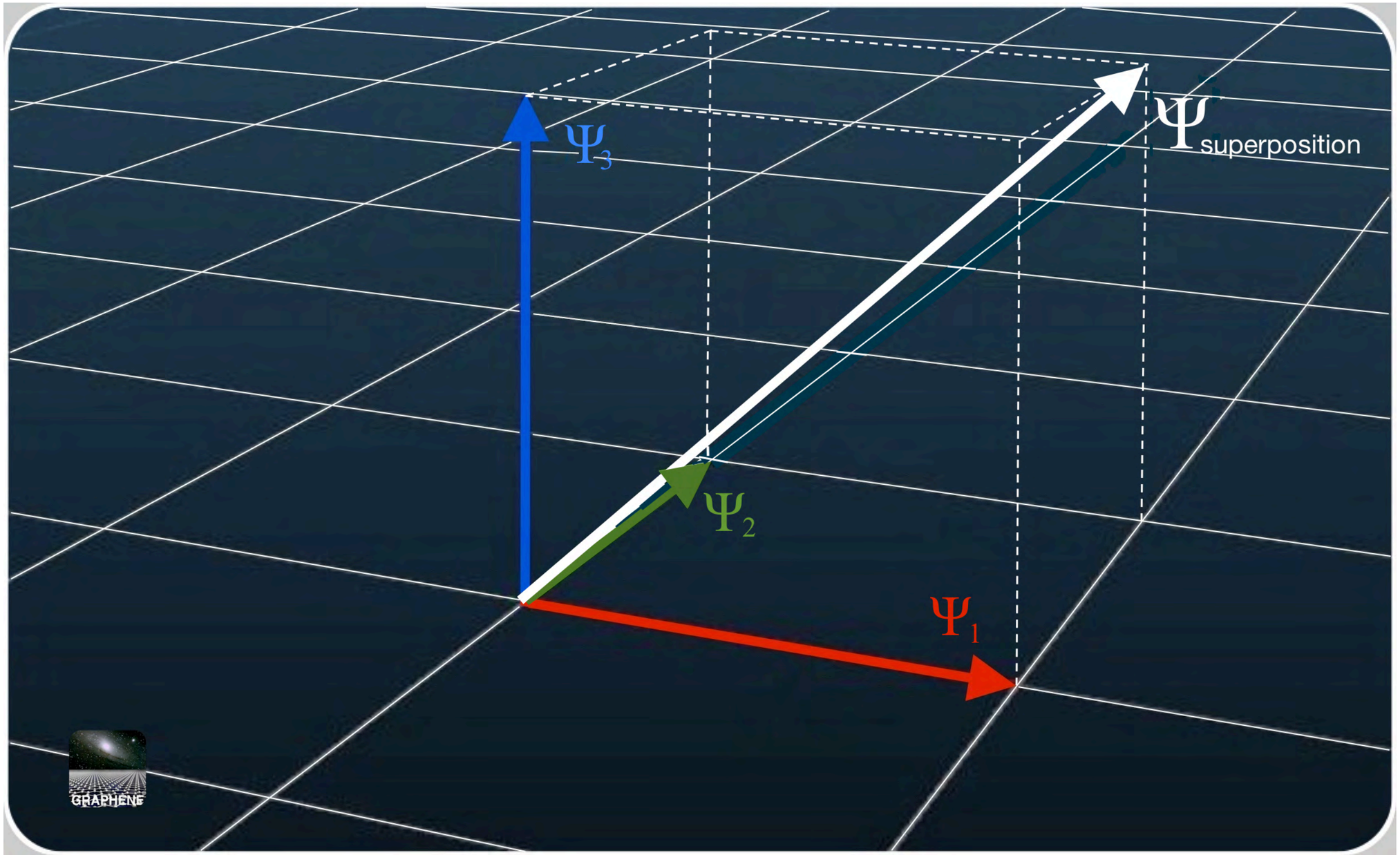
In [31]:
```
t = np.linspace(0, 1, samp_rate * len_in_sec)
sin_8hz = 1*np.sin(8 * 2 * np.pi * t)
sin_8_1hz = 1*np.sin(8.1 * 2 * np.pi * t)
np.dot(sin_8hz, sin_8_1hz)
```

Out[31]:
```
464.37702127330243
```

Out[28]: [<matplotlib.lines.Line2D at 0x10ba66320>]

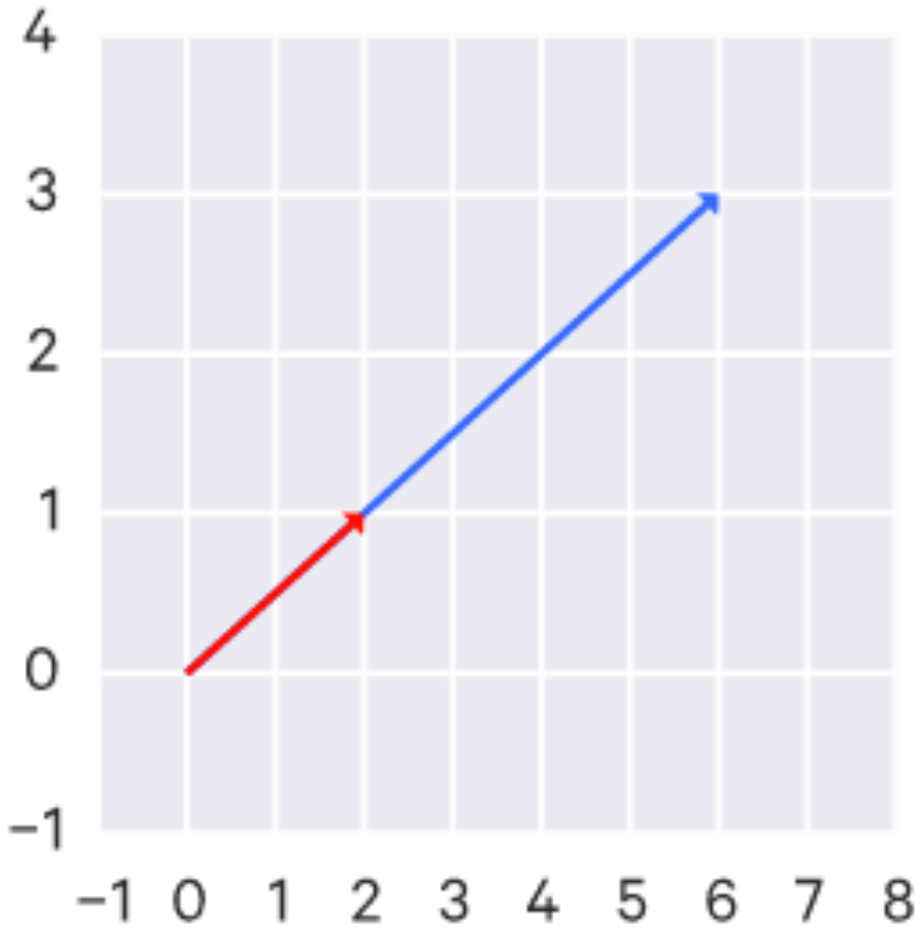


You can think of the set of all waves as forming Hilbert space where each frequency wave is a different dimension.

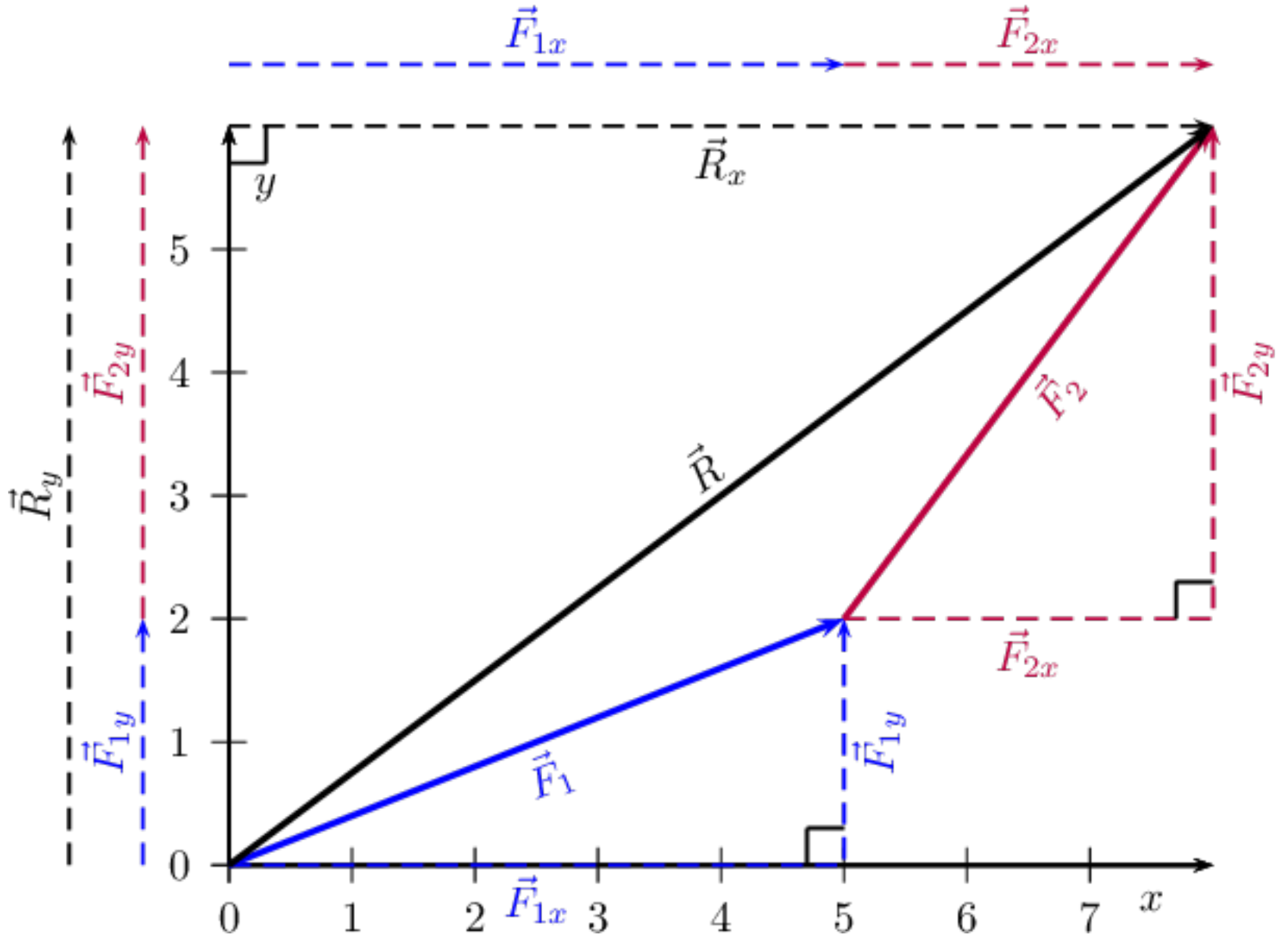


You can only deconvolute orthogonal components:

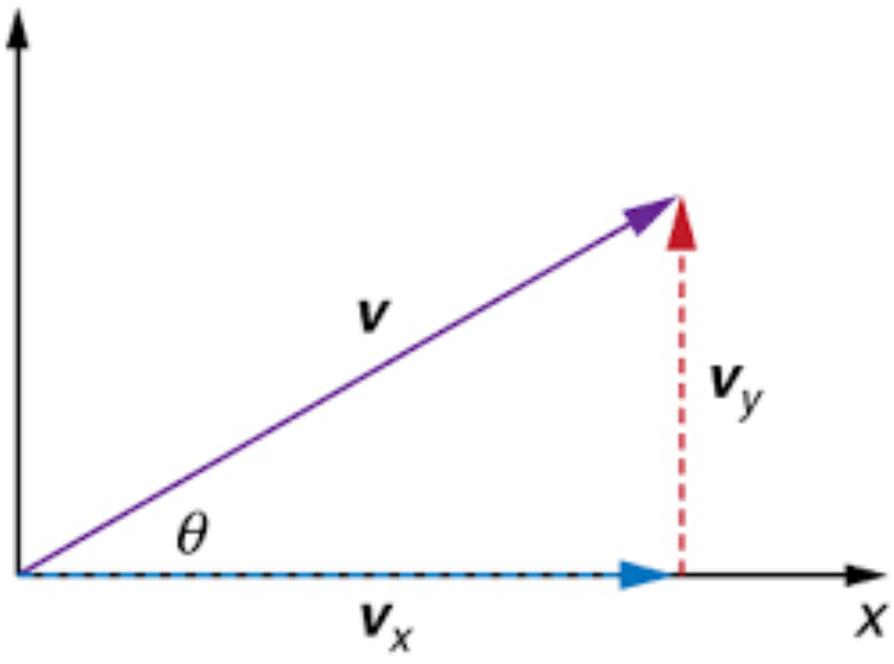
If you have 2 "components" in the same that are not orthogonal, you can't ever deconvolute them.



Even if they are not fully in the same dimension, if the vectors are not orthogonal, you can't fully recover the components:

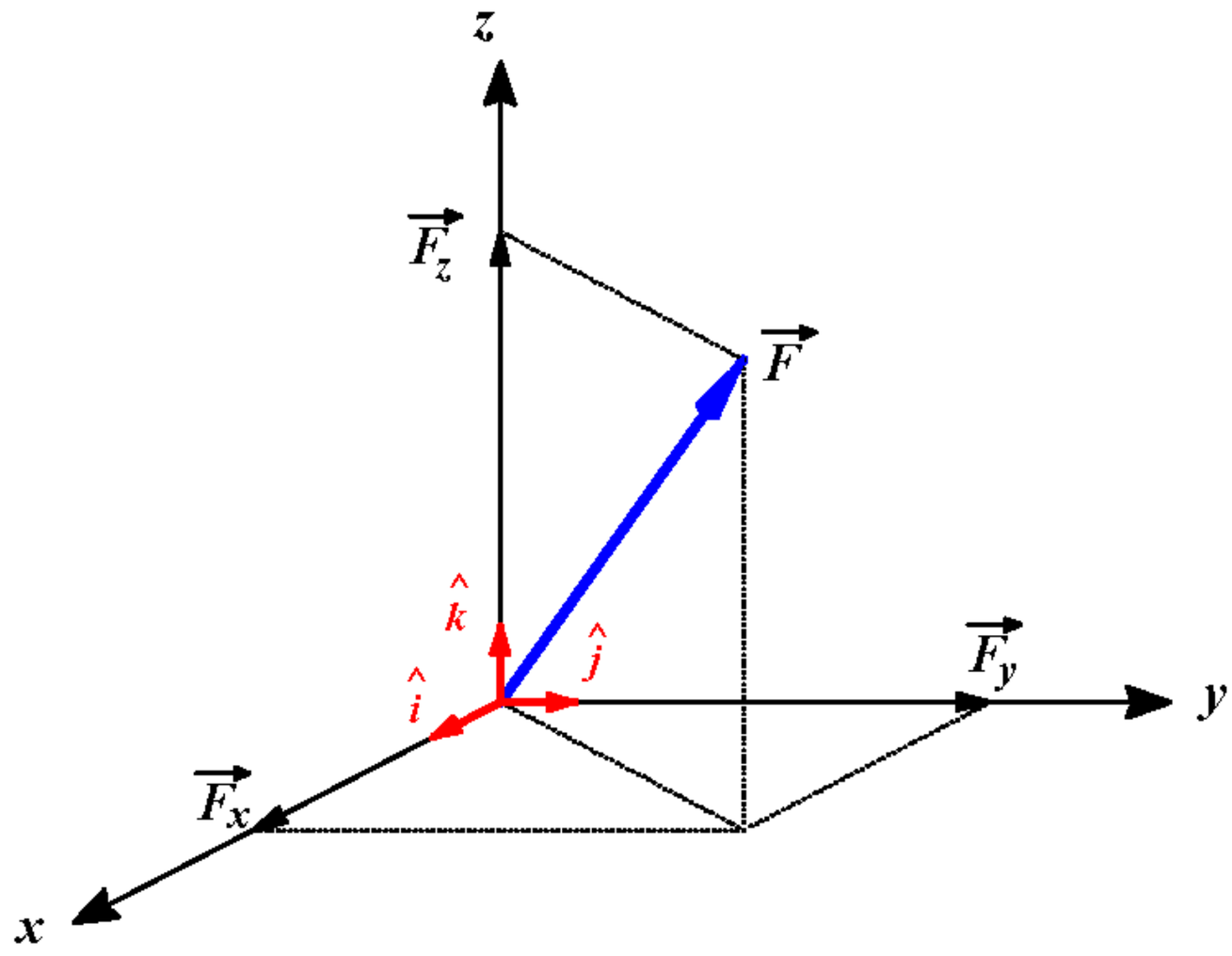


But you can fully recover the orthogonal components:



And that holds for higher dimensions too:

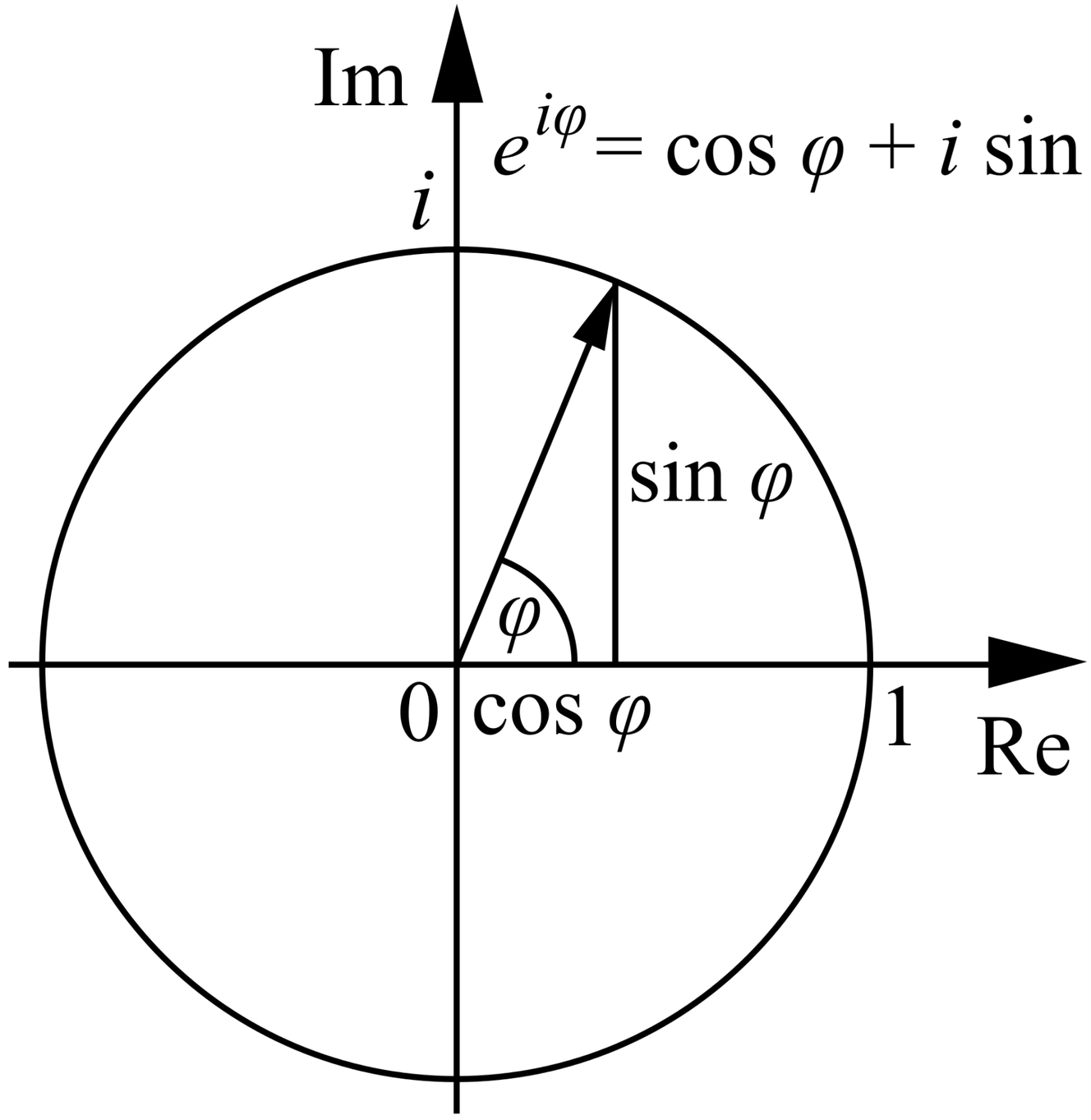
$$\vec{F} = \vec{F}_x + \vec{F}_y + \vec{F}_z$$



Relation to radio hacking: this is how, when you tune into a particular frequency, you don't get tons of interference from all the other waves in the air.

Relation to e

Euler's Formula relates e^x to sin(x) and cos(x):



This is also where we get Euler's Identity:

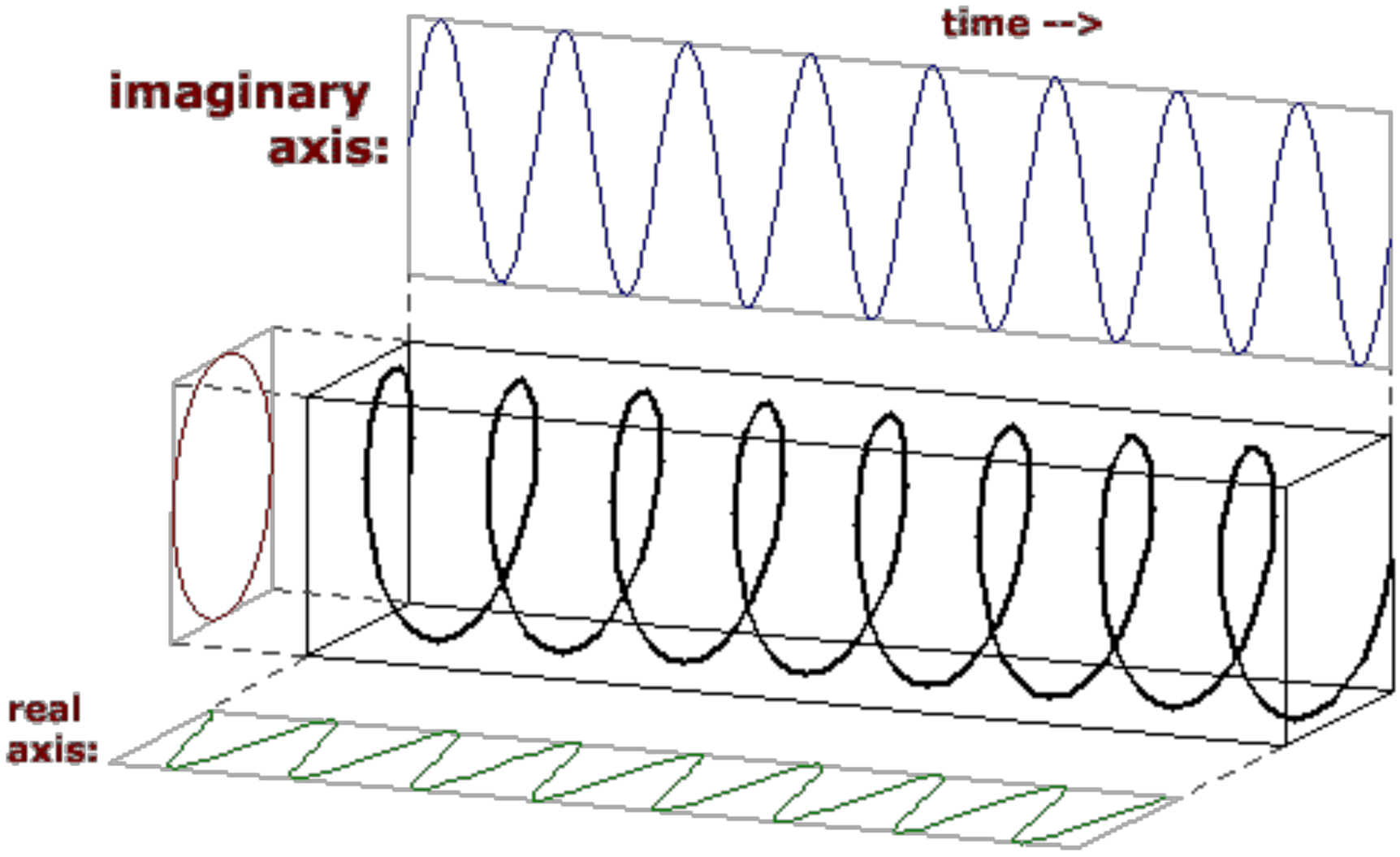
$$e^{i\pi}+1=0$$

At x = π, sin(π) = 0, and cos(π) = -1

So e^iπ = -1 + 0

Move the 1 over: e^iπ + 1 = 0

Complex sin waves



Relation to radio hacking: Radio waves are actually transmitted as complex waves, not simple waves, so if we want to craft our own radio waves to control things, we'll need to use complex sine waves.

How does digital communication happen over waves?

Modems

"Modem" = "**MO**dulator" + "**DEM**odulator"

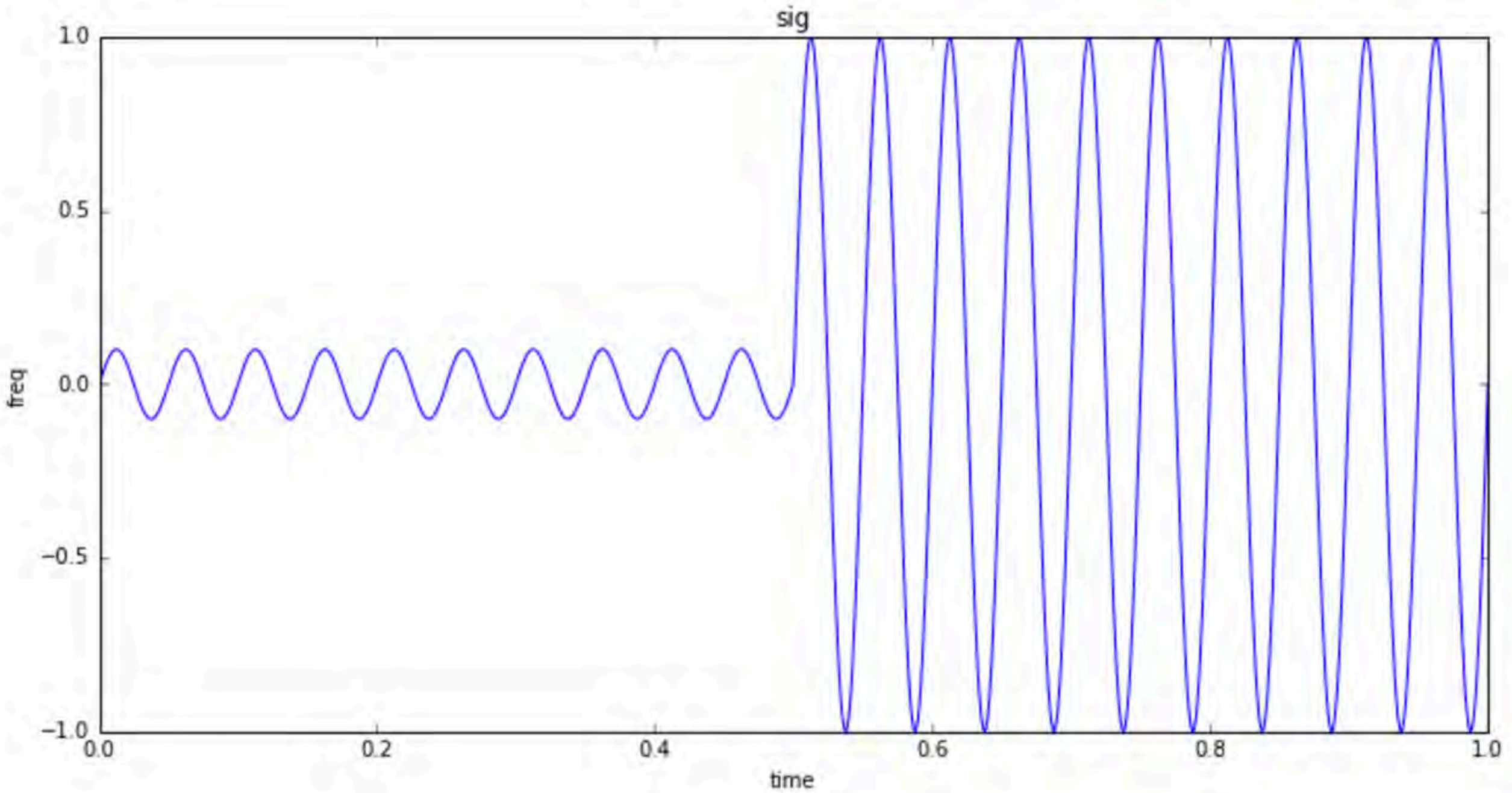
Modulation

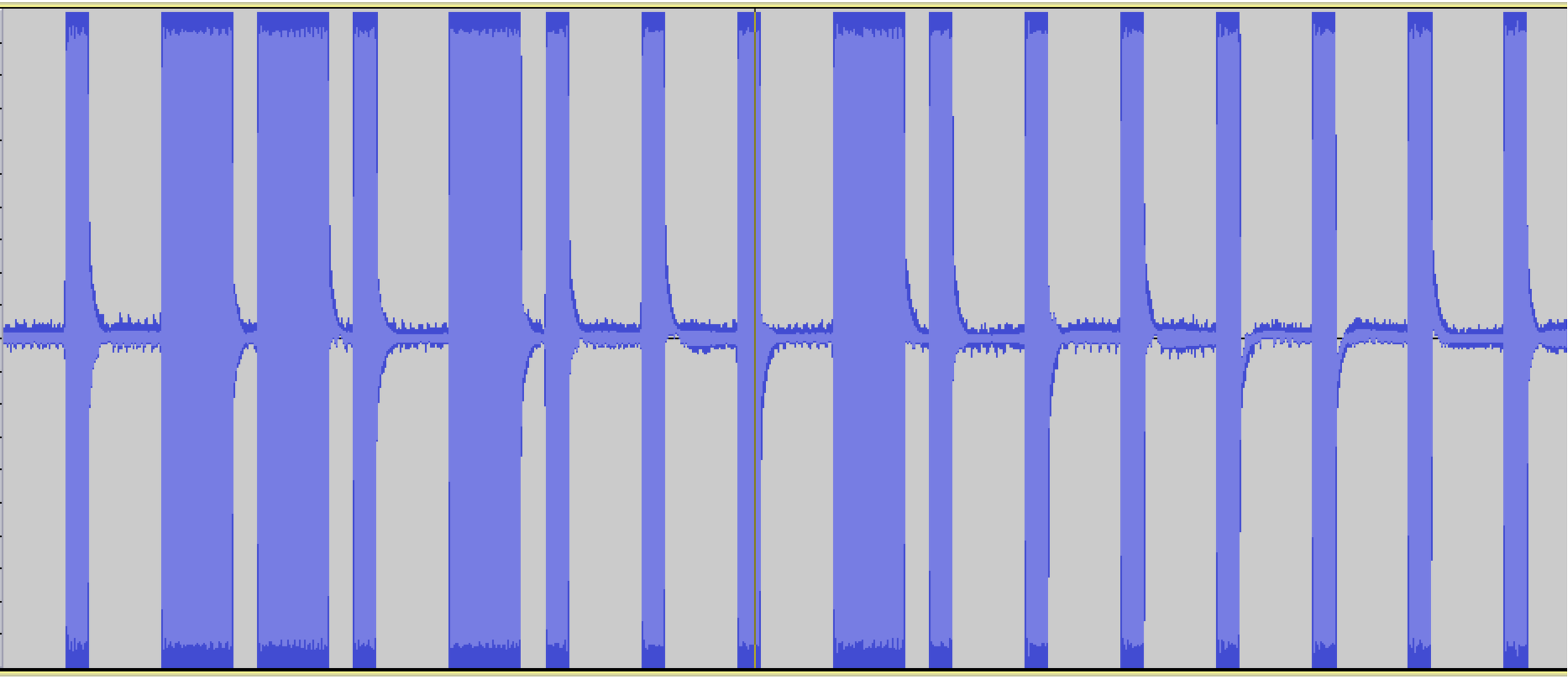
Modulation: manipulating a "carrier" wave to carry information.

Types of digital modulation:

- Amplitude-Shift Keying (ASK)
- Frequency-Shift Keying (FSK)
- Phase-Shift Keying (PSK)
- Quadrature Amplitude Modulation (QAM)

Amplitude-Shift Keying (ASK)

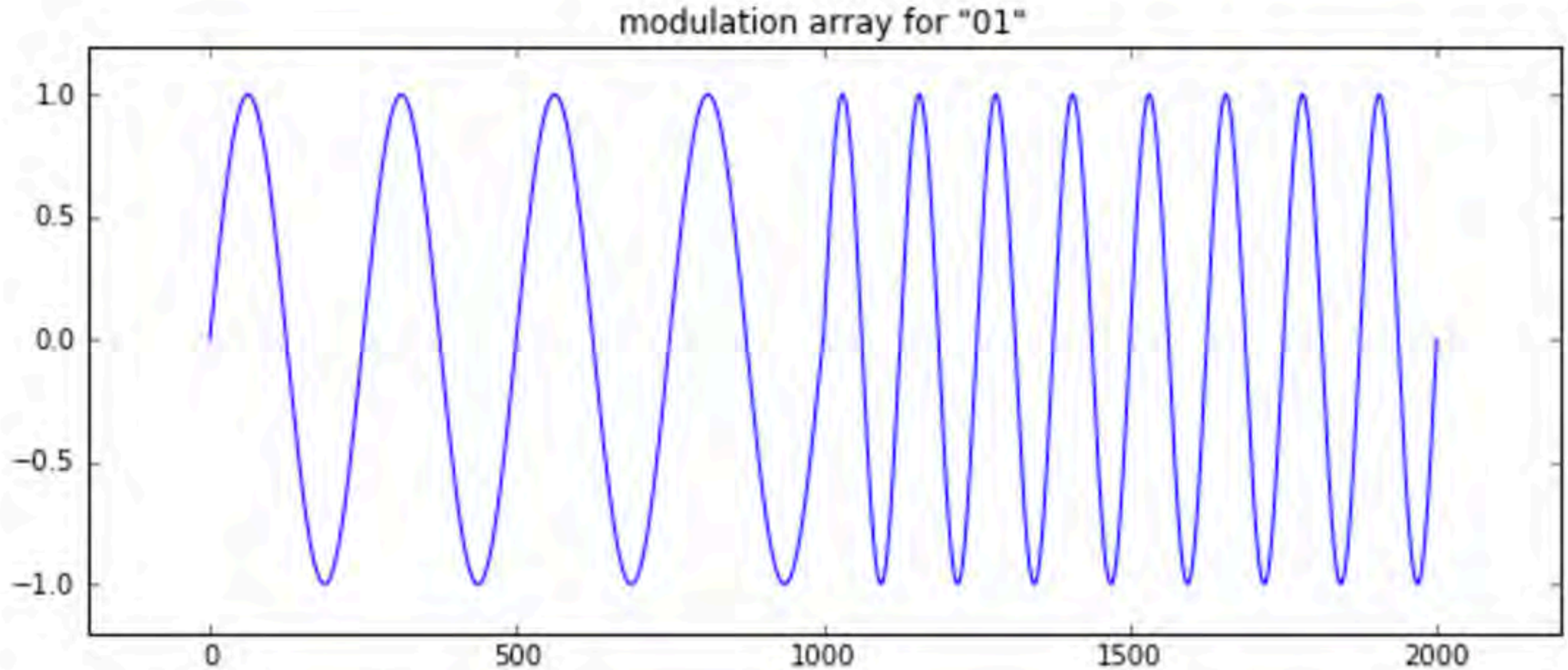


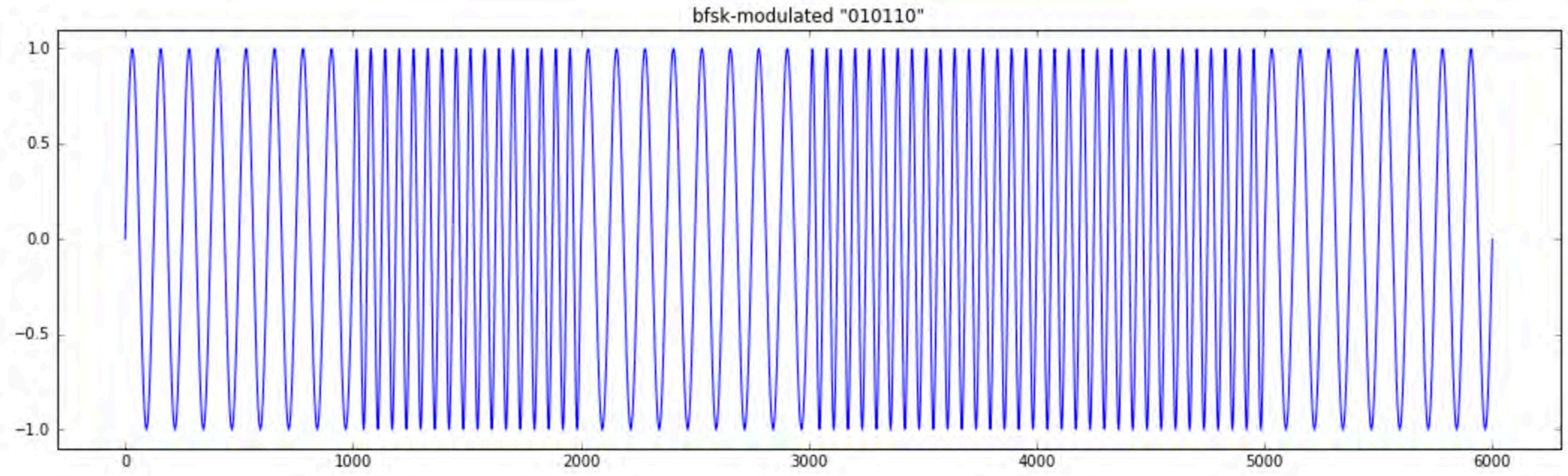


Used by many simpler devices

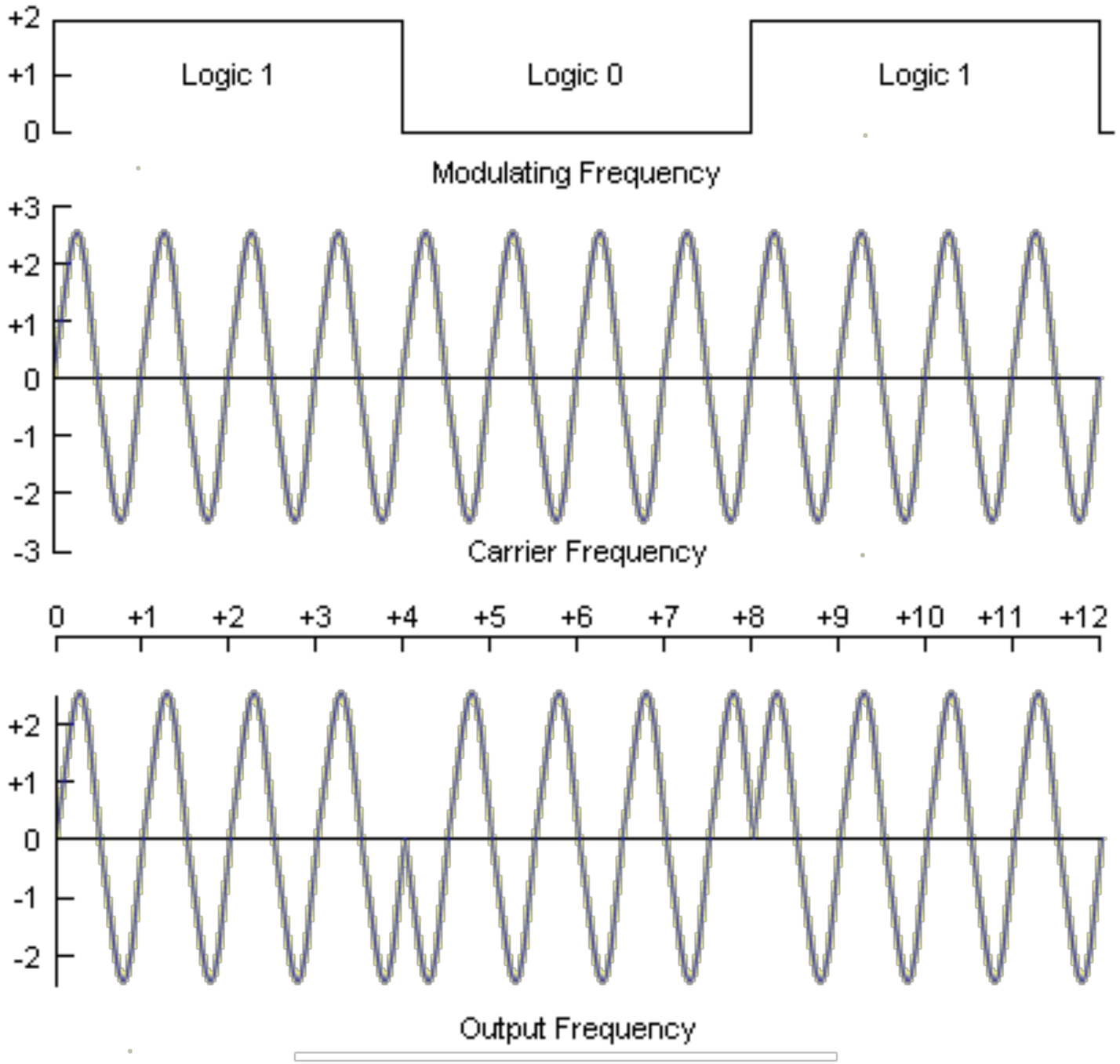
This is the type the wireless outlets I'm controlling uses

Frequency-Shift Keying (FSK)

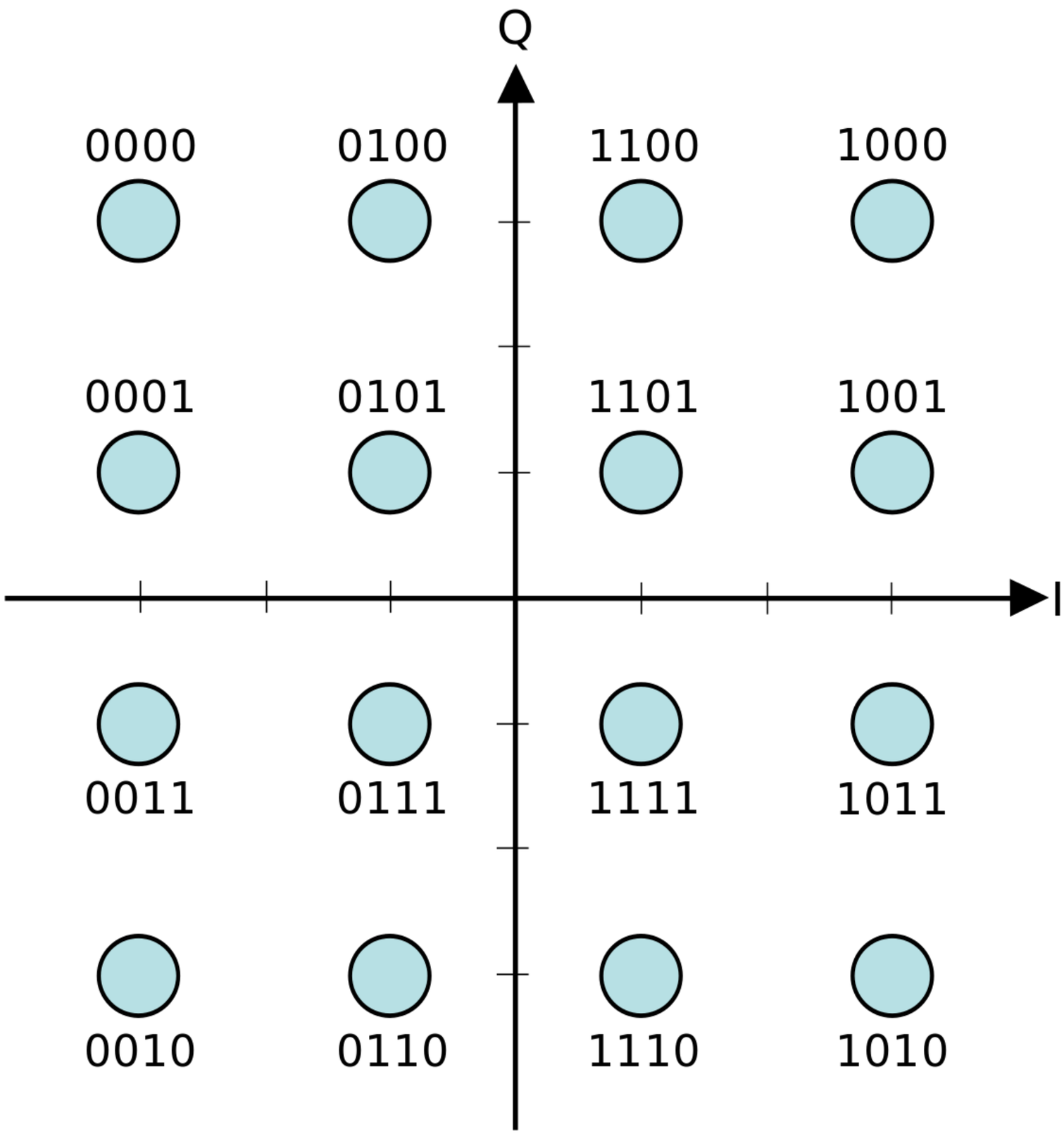



bfsk-modulated "010110"
1.0
0.5
0.0
−0.5
−1.0
0
1000
2000
3000
4000
5000
6000

Phase-Shift Keying (PSK)



Quadrature Amplitude Modulation (QAM)



Essentially, this is a combination of ASK and PSK

This is what modern WiFi mostly uses.

Actually generating a radio signal from scratch

Demo

- Jupyter Notebook
- Actually control the outlet

Script: https://github.com/calebmadrigal/radio-hacking-scripts/blob/master/ask_modulate_radio_signal.py

Jupyter Notebook: https://github.com/calebmadrigal/radio-hacking-scripts/blob/master/radio_signal_generation.ipynb

Conclusions

All wireless digital communications happens over EM waves.

And though some of the ways information is represented, all of it is still just plain ol' EM waves.

Now you know how the fundamentals of all radio communication happens!

Including some freakin' awesome foundational math.

Be aware of the potential attack vectors that all wireless systems:

Jamming attacks - not a lot to mitigate this type of attack; but and it's good to be aware of that. A few possible mitigation schemes:

Spread spectrum radio

Channel hopping

Active low - so if the signal goes away, consider that "triggered"

Replay attacks - make sure rolling codes are good

Brute force attacks - make sure key space is large enough and random enough

Mixed replay+brute force?

Be wary of wireless communication, and keep this stuff in mind when analyzing all the new IoT devices coming out!

Thanks!

Questions?

Caleb Madrigal
(Public) handle: metem
Website: http://calebmadrigal.com/
Twitter: @caleb_madrigal
Ham call sign: w0hak

Link to these presentation note: http://tiny.cc/hackwave
Link to code and jupyter notebooks: https://github.com/calebmadrigal/radio-hacking-scripts