SON 2026-2027 lecture 2

3TC Audio Project @ INSA Lyon

Tanguy Risset

https://inria-emeraude.github.io/son/

The Teensy Development Framework

Follow explanations of PJRC: https://www.pjrc.com/teensy/td_download.html

Download Arduino IDE (https://www.arduino.cc/en/software/), version 2.3.10

- e.g. For Linux: download ''arduino-ide_2.3.10_Linux_64bit.AppImage'',
- have it executable
- Once launched add the board manager URL as explained in PJRC's page (''https://www.pjrc.com/teensy/package_teensy_index.json'')
- Check the compilation of a simple example (e.e. "File -> Examples -> Teensy -> Tutorial1 -> Blink)
- In cas of problem, check the udev rules (see https://www.pjrc.com/teensy/td_download.html)

Break

wait until every on has a arduino running

clone the SON github

- clone the SON GitHub repository: https://github.com/inria-emeraude/son

  `git clone https://github.com/inria-emeraude/son somewhere-in-your-home`

- The github repository contains
  - In directory `examples/teensy/libraries/mydsp` , the code for the *mydsp* library used for basic audio programs on teensy.
  - In directory `examples/teensy/projects` , the arduino code of various basic audio projects.

Install the `mydsp` library in arduino

- From the SON github repository, copy the `examples/teensy/libraries/mydsp` into the folder `$ARDUINOPATH/libraries`
- where `$ARDUINO` can be seen in `File->Preferences->Scketchbook` location
- After that, the `$ARDUINOPATH/libraries` contains a directory named `mydsp`
- IMPORTANT: do not modify the programs of `mydsp/src` present in `$ARDUINOPATH/libraries` , copy them in your sketchbook directory if you want to modify them

Hello World

Write the following program in the programming interface:

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.println("Hello World!");
  delay(100);
}
```

Upload it on the Teensy (make sure that the right board and port are selected in `Tools`).

Open the serial debugger by clicking on the loop on the top right corner of the Arduino IDE.

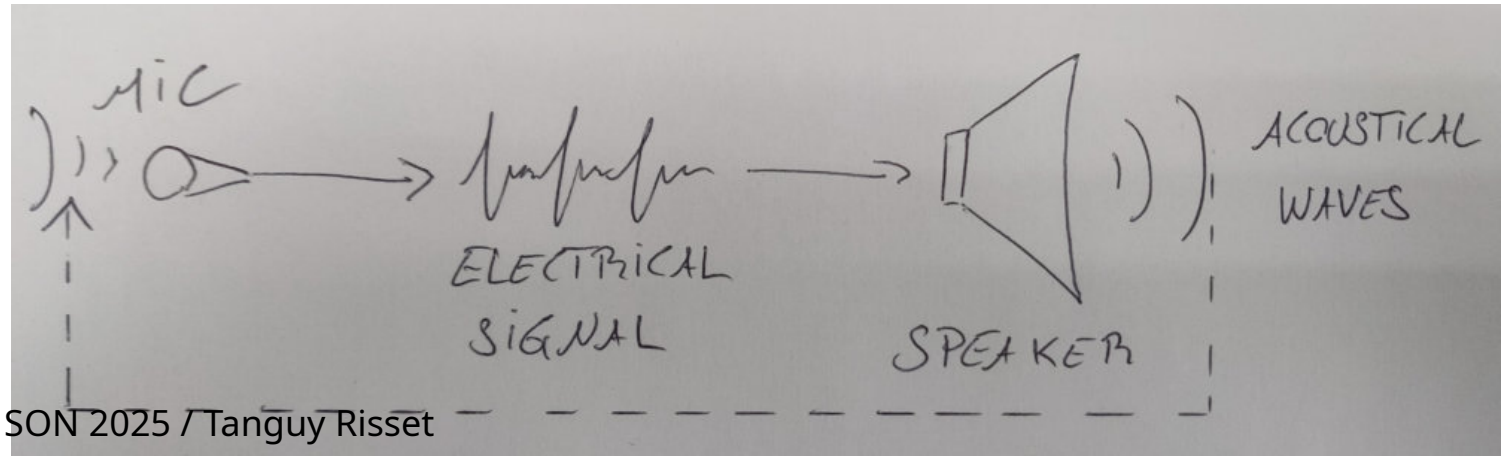
Running an Audio Program

- Open `examples/teensy/projects/crazy-sine` from your clone of https://github.com/inria-emeraude/son in the Arduino IDE
- Compile and run this program on your Teensy.
- Plug your headphones to the 3.5mm Jack connector on the Audio shield and check you hear the music.

Lecture 2, Part 2: Audio Signal Processing Fundamentals

Analog Audio Signals

- Analog signal is a wave, it ca be
  - Mecanical (transmitted through the air: sounds we hear)
  - Electrical (transmitted through wires)
- Conversion:
  - from elecrical to sound: loudspeakers
  - from sound to electrical: microphone



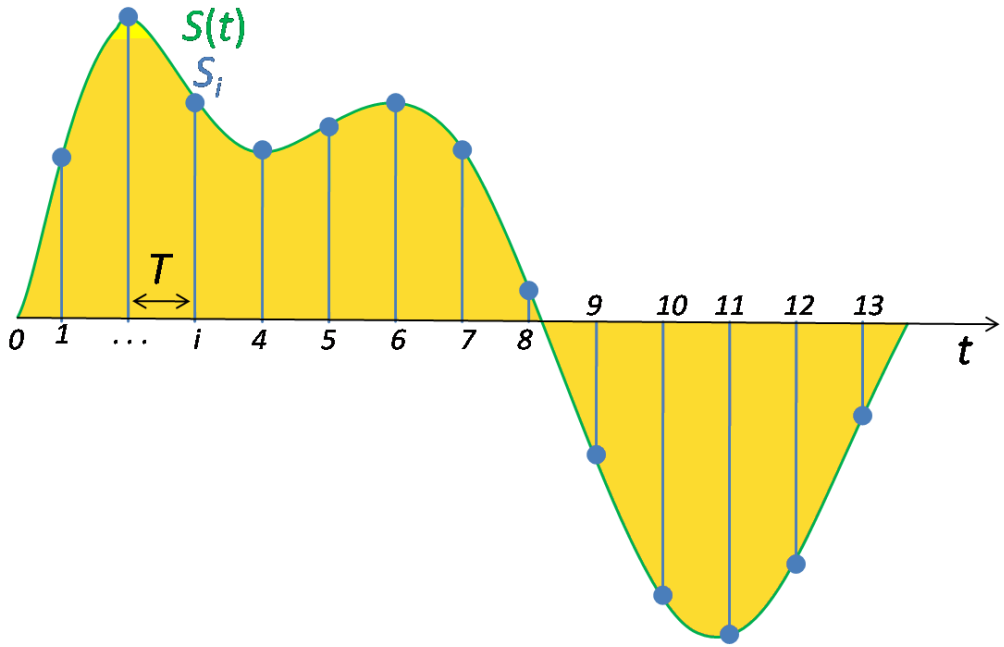
The Discovery of Digital Audio

- Sampling theory:
  - initial work by Harry Nyquist and was theorized in the 1930s by Claude Shannon to become the Nyquist-Shannon sampling theorem.
- Sampling in the field of audio:
  - voltage measurements are carried out at regular intervals of time on an analog electrical signal.
  - Each individual acquired value is called a *sample* and can be stored on a computer.

sampling audio signals

Signal sampling representation. The continuous signal is represented with a green colored line while the discrete samples are indicated by the blue vertical lines.(source: Wikipedia)

The number of samples per second also known as the sampling rate (noted $f_s = 1/T$), often 48kHz in audio systems.



ADC and DAC

In the field of audio, an ADC (Analog to Digital Converter) is a hardware component that can be used to discretize (sample) an electrical analog audio signal.

The reverse operation is carried out using a DAC (Digital to Analog Converter). In most systems, the ADC and the DAC are hosted in the same piece of hardware called *audio codec* or *audio interface*.

Human hearing

- Understanding human hearing is a mix of physical law and physiological issues (domain called "Perception": ear and brain work a lot when we hear)
- In theory, humans can hear any sound between 20 and 20000 Hz.
- In practice, most adults can't hear frequencies over 17 kHz.

Nyquist Frequency

- When sampling an audio signal, the sampling rate (f_s) will determine the highest frequency than can be sampled by the system (f_n) also known as the "**Nyquist Frequency**".

$$f_n = \frac{f_s}{2}$$

- The highest frequency that can be sampled is half the sampling rate.
- Hence, in order to sample a frequency of 20 kHz, the sampling rate of the system must be at least 40 kHz (40000 samples per second).
- The standard for modern audio systems is to use a sampling rate of 48 kHz. f_s is 44.1 kHz on compact discs (CDs) and many home and recording studios use a sampling rate of 96 or 192 kHz.

Sampling Theorem (Shannon)

Let $x(t)$ denote any continuous-time signal having a continuous Fourier transform:

$$X(f) \triangleq \int_{-\infty}^{\infty} x(t) e^{-2i\pi ft} dt$$

Let

$$x_d(n) \triangleq x(nT), \quad n = \ldots, -2, -1, 0, 1, 2, \ldots,$$

denote the samples of $x(t)$ at uniform intervals of T seconds. Then $x(t)$ can be exactly reconstructed from its samples $x_d(n)$ if $X(f) = 0$ for all $|f| \geq /2T = f_s$.

In other words, any frequency (harmonics) between 0 Hz and the Nyquist frequency can be **exactly reconstructed** without loosing any information.

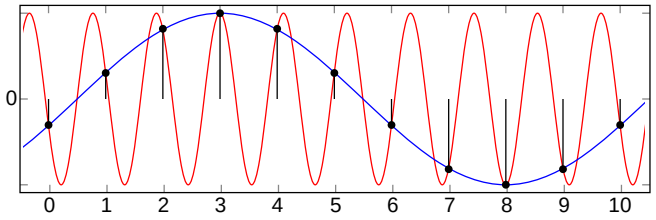
Aliasing

In audio, aliasing happens when a digital signal contains frequencies above the Nyquist frequency.

For all frequency f_o above fn, the sampled frequency f will be:

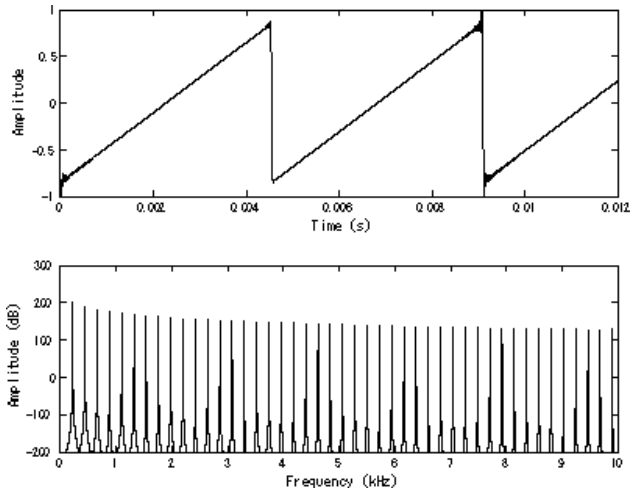
$$f = f_n - (f_o - f_n)$$

Example (Wikipedia): $f_o = 0,9$, confused with a signal of frequency $f = 0,1$ for a sampling with a period $T = 1,0$ (i.e $f_n = 0,5$)



Preventing Aliasing

- During Sampling Aliasing is prevented by filtering an analog signal before it is discretized by removing all frequency above fn.
- Aliasing can also be obtained when synthesizing a broadband signal on a computer (e.g., a sawtooth wave). It is the software engineer's role to prevent this from happening.



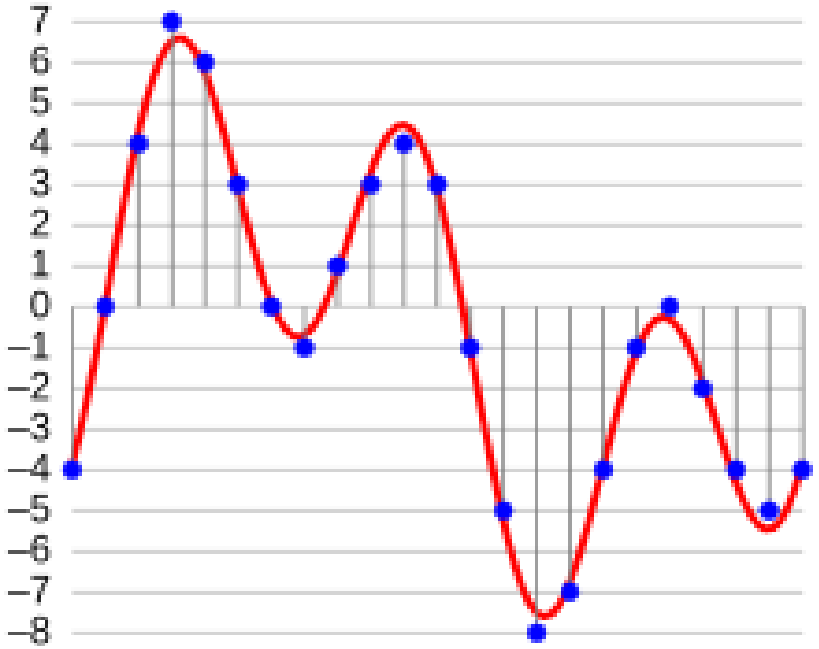
Sawtooth wave at 220 Hz and its spectrum (Wikipedia)

Bit Depth, Dynamic Range

- Audio is typically recorded at 8, 16 (the standard for CDs), or 24 bits (and 32 bits in some rarer cases)
- The established standard in audio is that audio signals coded on decimal numbers always have the following range: {-1;1}
- For example, if audio samples are coded on 16 bits unsigned integers,
  - the value s_{int} range for $-2^{15} = -32768$ to $-2^{15} - 1 = 32767$
  - the "real" value intepreted for signal s is $s_{int}/32768$ which rage from -1 to 1.
- In software, samples can be processed in floating point, double or fixed point
- Most systems will clip audio signals to this range
- But ADC will convert to 8, 16, 24 or 32 bit fixed point integers

Bit Depth and Signal-to-Noise Ratio

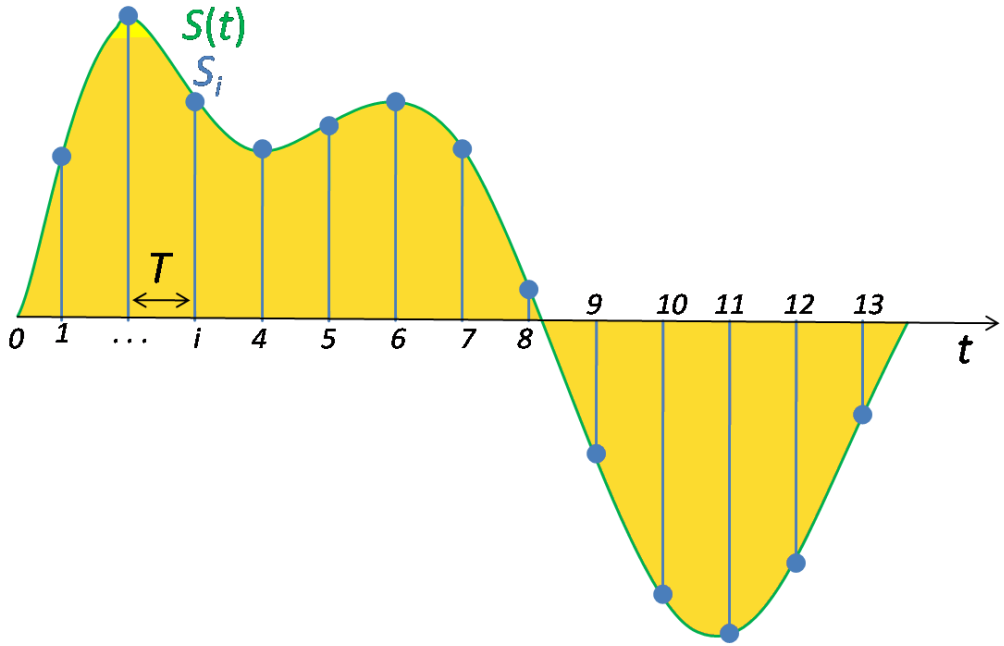
- Bit depth is the number of bits of information in each sample
- It corresponds to the resolution of each sample
- it depth primarily affect the noise level from quantization error—thus the signal-to-noise ratio (SNR)
- techniques such as dithering, noise shaping, and oversampling can mitigate these
- Audio SNR is 146 dB for 24 bits, 98 dB for 16 bits and 25 dB for 4 bits



An analog signal (in red) encoded to 4-bit PCM digital samples (in blue) (Wikipedia)

Summary

- When Sampling a signal
  - A sinusoid frequency determines the pitch
  - (most of the time the signal is composed of a huge number of sinosoides)
  - The range of the signal determines the volume
  - sample rate limits aliasing
  - bit depth limits SNR
  -



End of Lecture 2