

# Spectrum Analyzer

Lavoro Professionale Individuale

---

Naoki Pross

25 maggio 2018

SAM Bellinzona

# Table of contents

1. Introduzione
2. Fourier Transform
3. Fast Fourier Transform
4. Prodotto realizzato
5. Conclusioni

# Introduzione

---

# A cosa servono l'analisi spettrale e la FT?

Alcuni esempi pratici

**Elettronica** Filtri digitali, analisi di segnali

**Informatica** Audio editing, Riconoscimento sonoro / vocale

**Matematica** Risoluzione di equazioni differenziali

**Fisica** Principio di indeterminazione di Heisenberg



$$\langle x | \hat{p} | \psi \rangle = -i\hbar \frac{d}{dx} \psi(x)$$

$$\langle x | \hat{x} | \psi \rangle = i\hbar \frac{d}{dp} \psi(p)$$

Realizzare un circuito di analisi spettrale

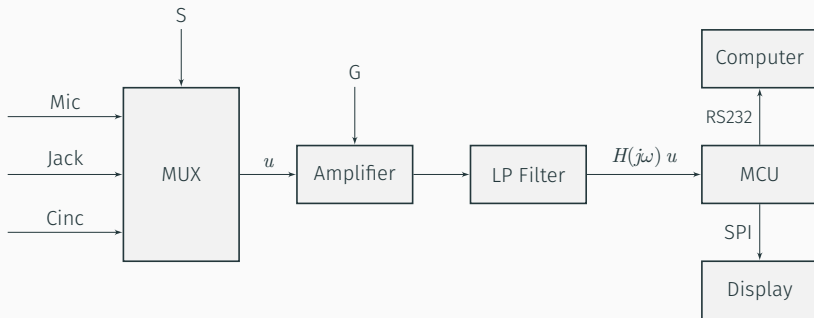
## Requisiti

- Analisi dello spettro fino a 10 kHz
- Entrate Jack e RCA
- Visualizzazione
- Utilizzo di un PIC18F45K22

## Componenti

- Circuito di adattamento in entrata
- Design di un PCB
- Software per il uC e per il PC

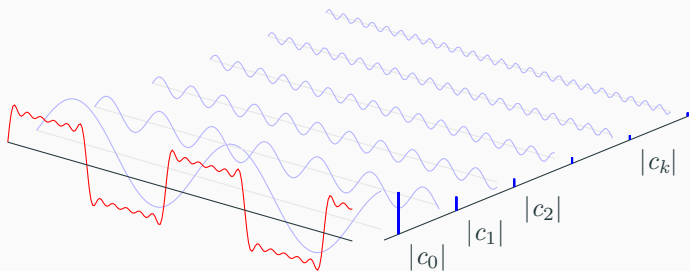
# Schema a blocchi



# Fourier Transform

---

# Rappresentazione grafica



$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(x) \cdot e^{-i\omega x} dx$$



# Fast Fourier Transform

---

# Divide et impera (Divide and Conquer)

La FFT appartiene ad una classe di algoritmi chiamata

## Divide et Impera

### Definizione

1. *Divide* il problema in sottoproblemi
2. *Impera* (conquista) i sottoproblemi
3. *Combina* i risultati dei sottoproblemi

# Il problema della DFT

La trasformata di Fourier discreta è

$$\vec{X} = \mathbf{V} \cdot \vec{x}_n$$

In cui

$$\mathbf{V} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & (e^{i\omega})^{-1} & (e^{i\omega})^{-2} & \dots & (e^{i\omega})^{-(N-1)} \\ 1 & (e^{i\omega})^{-2} & (e^{i\omega})^{-4} & \dots & (e^{i\omega})^{-2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & (e^{i\omega})^{-(N-1)} & (e^{i\omega})^{-2(N-1)} & \dots & (e^{i\omega})^{-(N-1)^2} \end{pmatrix}$$

# Complessità temporale (analisi asintotica)

## Definizione

In informatica, la complessità temporale di un algoritmo quantifica la quantità di tempo impiegata da un algoritmo a essere eseguito.

Descrivendo il prodotto nelle componenti

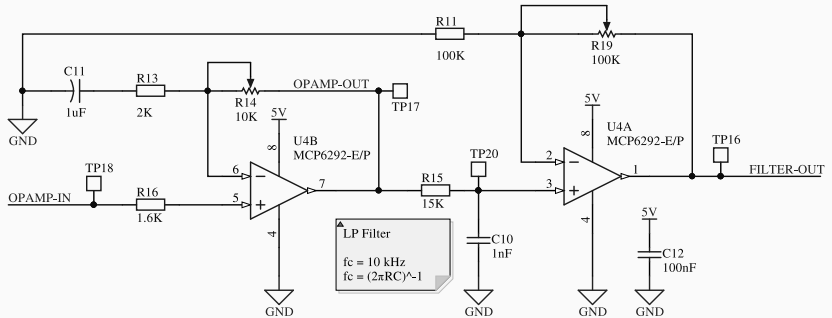
$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i\omega kn} \quad 0 \leq k < N$$

Osserviamo che la complessità temporale è  $\mathcal{O}(N^2)$ .

Prodotto realizzato

---

## Signal Adapter



L'implementazione si chiama

## Fixed-Point In-Place FFT

**Fixed-Point** Il numero di bit per la mantissa e per l'esponente del valore floating point sono costanti

**In-Place** I risultati sovrascrivono i dati di partenza

**FFT** Fast Fourier Transform

Demo!



# Conclusioni

---

## Raggiungi

- Analisi spettrale fino a 10 kHz
- Visualizzazione al PC<sup>†</sup>
- Esportare immagini / dati

## Incompleti

- <sup>†</sup> Visualizzazione delle curve  $\Re(z)$  e  $\Im(z)$  in un solo grafico
- Visualizzazione mediante la matrice LED

