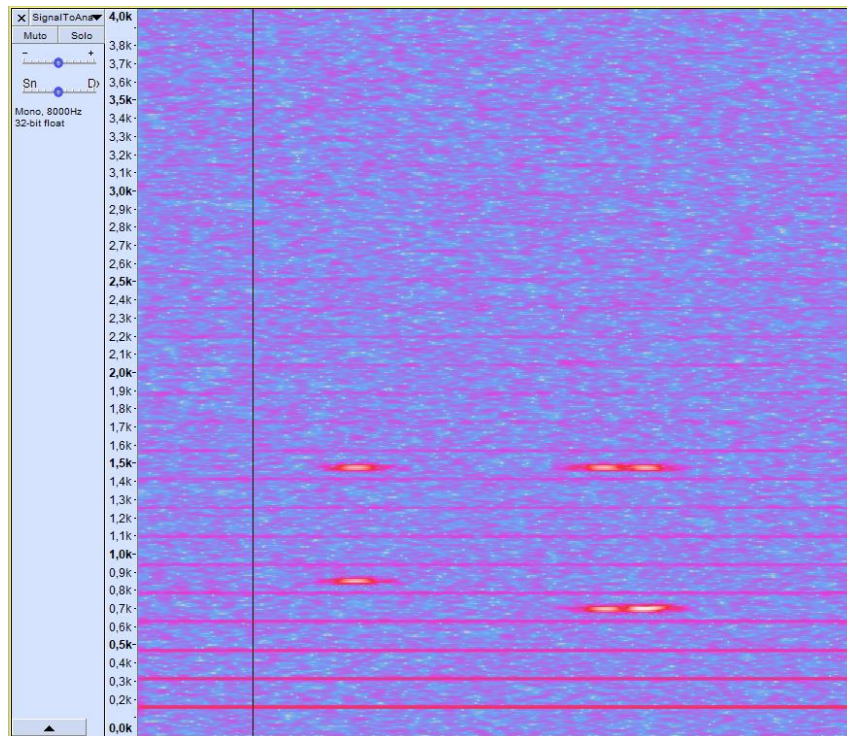




Per risolvere il problema per prima cosa bisogna identificare la frequenza di ogni singolo tasto, quindi eliminiamo il rumore dalla traccia ([https://www.youtube.com/watch?v=iZCTuzrRU\\_Y](https://www.youtube.com/watch?v=iZCTuzrRU_Y)) e successivamente è necessario ricevere lo spettrogramma (cambiando da forma d'onda a spettro) con Audacity (<https://www.youtube.com/watch?v=7WYw3qoTdU4>). Il primo passo è facoltativo siccome non è necessario ma caldamente consigliato per un'analisi visiva (siccome si vede meglio).



Successivamente prendiamo la tabella che si trova su [https://en.wikipedia.org/wiki/Telephone\\_keypad](https://en.wikipedia.org/wiki/Telephone_keypad) per poter analizzare ogni singolo tasto. Ad esempio vediamo che il primo tasto deve essere un 9 perché si trova in 1477 Hz e 852 Hz.

**DTMF keypad frequencies (with sound clips)**

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

**La struttura dell'applicazione che intendereste sviluppare, con dettagli sugli strumenti e tecniche da utilizzare?**

Per poter creare un'applicazione bisognerebbe utilizzare un algoritmo di Short-Time Fourier Transform (STFT). In pratica bisogna sovrapporre più STFT e sommare ogni singolo punto in base al tempo.

[https://github.com/pydanny/pydanny-event-notes/blob/master/Pycon2008/intro\\_to\\_numpy/files/pycon\\_demos/windowed\\_fft/short\\_time\\_fft\\_solution.py](https://github.com/pydanny/pydanny-event-notes/blob/master/Pycon2008/intro_to_numpy/files/pycon_demos/windowed_fft/short_time_fft_solution.py)

Infine prendere i picchi per ogni istante di tempo e confrontarli con la tabella presente su Wikipedia (Telephone Keypad). Per ricevere poi il messaggio in formato testuale è necessario "simulare" il formato di scrittura T9. Fonte <https://stackoverflow.com/questions/4431481/frequency-detection-from-a-sound-file>