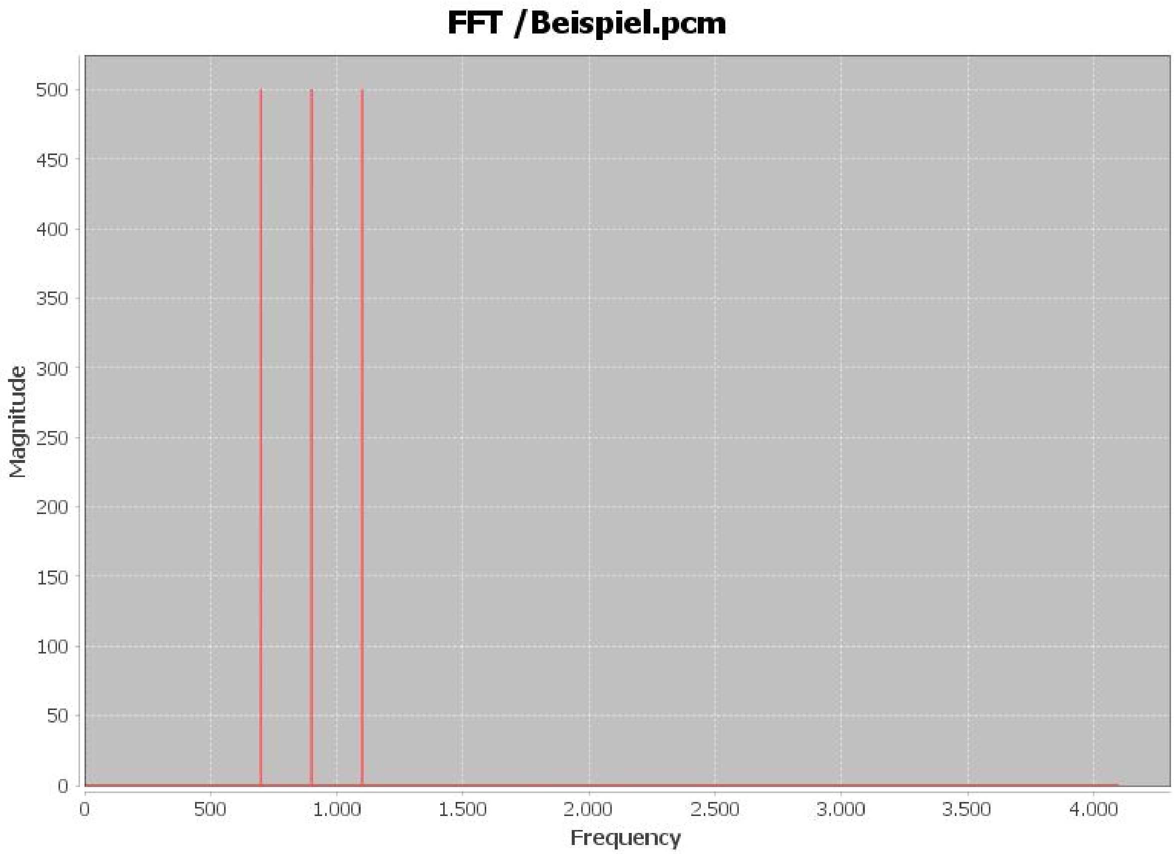
Andreas Braun (1200197)
Martin Braun (1249080)

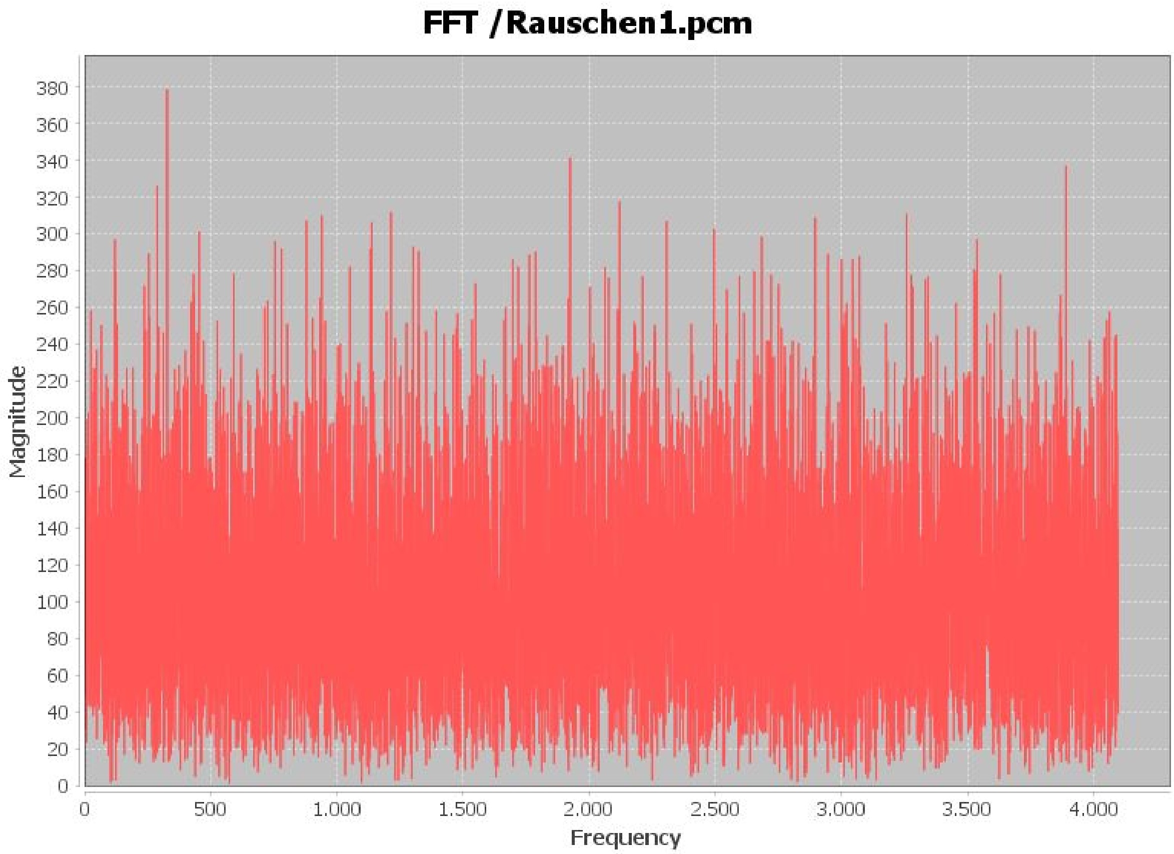
1a)



Die Frequenzen 700Hz, 900Hz, 1100Hz sind zu erkennen. Der Phasenversatz zwischen 700Hz und 1100 Hz beträgt |-0.1666 -0.4999| = 0.333 rad

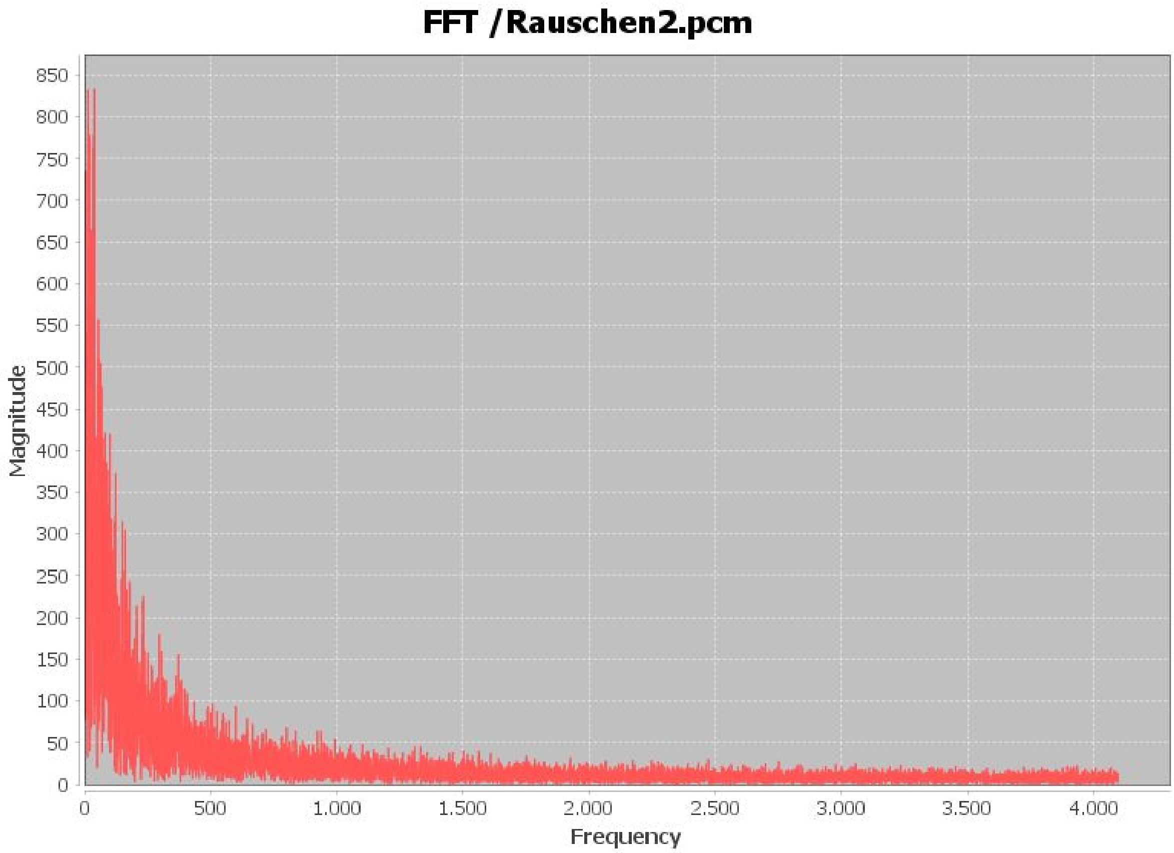
Andreas Braun (1200197)
Martin Braun (1249080)

1b)



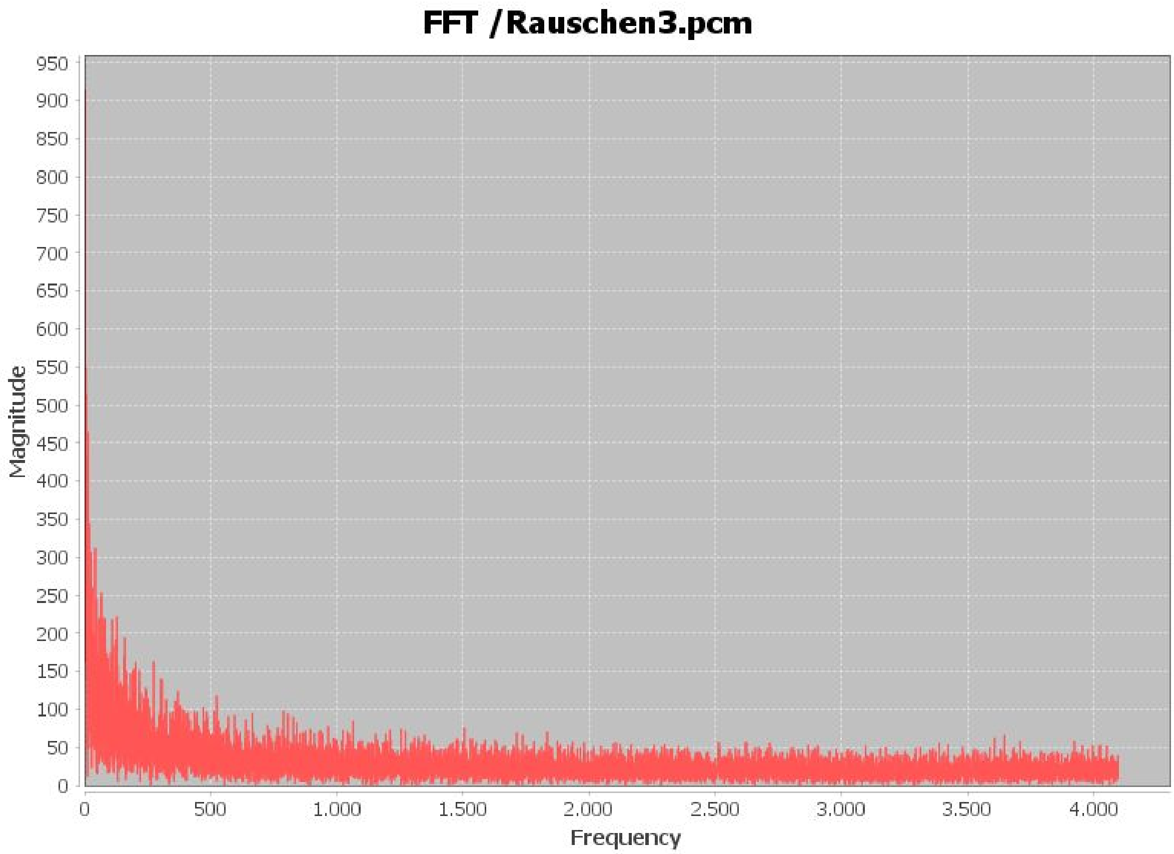
Rauschen1.pcm zeigt weißes Rauschen. “Gleich viel” Rauschen ist in jedem Frequenzspektrum zu erkennen.

Andreas Braun (1200197)
Martin Braun (1249080)



Rauschen2.pcm zeigt rotes Rauschen. Das ist erkennbar am großen Amplitudenunterschied des Bereichs 0-500Hz im Vergleich zu >500Hz des Rauschens.

Andreas Braun (1200197)
Martin Braun (1249080)



Rauschen3.pcm zeigt rotes Rauschen. Das ist erkennbar am geringer ausfallenden Abfall in der Amplitude vom Bereich 0-500Hz zum hochfrequenteren Bereich >500Hz.

1c)
Das Programm erkennt die vier 2-er Frequenztupel:

[[855.0, 1339.0], [773.0, 1339.0], [700.0, 1212.0], [855.0, 1480.0]]

Zuordnung:

DTMF keypad frequencies				
	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

https://en.wikipedia.org/wiki/Dual-tone_multi-frequency_signaling

Also ist die Telefonnummer (Die Frequenzen von oben werden zu der nächsten in der Tabelle zugeordnet):

8 5 1 9

Das Programm separiert das Signal einfach in 8 Teile und führt dann auf diesen die FFT aus. Da alle Töne unterschiedlich sind (oder zumindest keine zwei gleiche aufeinander folgen) können wir Wechsel zwischen den Tönen erkennen: diese haben mehr als 2 “dominante” Frequenzen.

In der .zip Datei sind die FFTs der Teilsignale zu sehen

Andreas Braun (1200197)
Martin Braun (1249080)

1 d) `double frequency = (double)i * samples_per_second / N;`

N fließt in die Fouriertransformation über das Array mit ein. Wenn wir ein Signal bzw. eine Datenreihe der Länge N in die FFT übergeben, bekommen wir auch wieder eine Datenreihe der Länge N zurück. Nur dass das Signal jetzt im Frequenzbereich angesiedelt ist (vorher im Zeitbereich).

In der for Schleife um diesen Ausdruck wird zudem i in den Bereich von 0 bis N / 2 - 1 eingegrenzt (das folgt aus dem Abtasttheorem: Wir haben FFT für N Werte berechnet, also sind im allgemeinen nur die Werte von 0 bis N / 2 -1 für die Frequenzanalyse interessant). Durch das Sampling müssen wir hier noch mit der Samplingfrequenz (in unserem Fall 8192Hz) multiplizieren bevor wir durch N teilen, da die FFT ja unabhängig von der Abtastfrequenz berechnet wurde. (Wenn z.B. N = 2 * samples_per_second, dann entsprechen zwei aufeinanderfolgende Einträge des Arrays derselben Frequenz).

1e)

Da die größte zu erkennende Frequenz 1633Hz entspricht, folgt aus dem Abtasttheorem eine minimale Abtastfrequenz von 1633Hz * 2 = 3266Hz. Würde ein geringer Abtastfrequenz verwendet werden, so würde eine Unterabtastung stattfinden analog wie auf 2.4-13. Das delta_t, mit dem abgetastet wird, ist zu groß und bei der Rekonstruktion des Signales aus dem abgetastetem Signal entsteht ein verfälschtes Signal.