

Telekomunikacja - laboratorium			Studia stacjonarne - inżynierskie		
Nazwa zadania		Ćwiczenie nr 4. Przetwarzanie analogowo-cyfrowe A/C			
Dzień	poniedziałek	Godzina	14:00-15:30	Rok akademicki	2020/2021
Imię i Nazwisko	Adam Kapuściński 229907				
Imię i Nazwisko	Damian Szczeciński 230016				
Imię i Nazwisko	-				
Opis programu, rozwiązania problemu.					
<p>Celem zadania było zbudowanie modułu przetwarzania A/C i C/A realizującego konwersję dźwięku z postaci analogowej do postaci cyfrowej i odwrotnie. Realizowany za pomocą karty dźwiękowej. Dźwięk płynący do karty dźwiękowej z mikrofonu lub innego urządzenia zostaje poddany przetwarzaniu A/C i przesłany w postaci strumienia do dalszego przetwarzania. W programie zaimplementowaliśmy 3 poziomy kwantyzacji i możliwość zmiany częstotliwości próbkowania. Według zaleceń również zamieściliśmy w naszym programie funkcję odpowiadającą za obliczanie wartości współczynnika SNR.</p> <p>Przetwarzanie analogowo cyfrowe jest to przetwarzanie sygnału analogowego na sygnał cyfrowy czyli zrozumieliśmy dla jednostek komputerowych. W skład przetwarzania wchodzi 3 podstawowe składniki: kwantyzacja, próbkowanie i kodowanie.</p> <p>Próbkowanie – jest to wyznaczanie momentów pomiarów wartości napięcia elektrycznego podawanego na wejściu przetwornika A/C w taki sposób, aby otrzymany ciąg liczb możliwie wiernie odwzorowywał zmiany tego sygnału.</p> <p>Kwantyzacja – jest to przedstawienie mierzonej wartości jako liczby elementarnych kwantów (u nas w układzie dwójkowym)</p> <p>Kodowanie – kodowanie otrzymanych wartości na etapie kwantyzacji do postaci dziesiętnej lub do kodu najodpowiedniejszego do przesyłania na przykład kodu z kontrolą parzystości lub kodu korekcyjnego.</p>					
Najważniejsze elementy kodu programu z opisem.					
Przetwarzanie A/C					
<pre>1. class Recorder: 2. 3. def __init__(self, device = None, samplerate : Optional[int] = 48000, channels : Optional[list] = [0,1]) -> None: 4. """ 5. device : mikrofon z biblioteki soundcard (domyślnie, domyślny)\n 6. samplerate : częstotliwość próbkowania\n 7. channels : lista kanałów na których ma być nagrywany dźwięk 8. """ 9. 10. self.device = device if device!=None else sc.default_microphone() 11. 12. self.samplerate = samplerate 13. 14. self.channels = channels 15. 16. self.audiosave = None 17. 18. 19. def record(self, seconds : int) -> None: 20. """ 21. seconds - ile sekund nagrać 22. """ 23. 24. # jeżeli self.audiosave jest puste to przypisz do niego nagranie 25. if type(self.audiosave) != np.ndarray: 26. self.audiosave = self.device.record(int(seconds * self.samplerate), self.samplerate, self.channels) 27. # jeżeli self.audiosave nie jest puste dopisz do niego nagranie 28. else: 29. test = self.device.record(int(seconds * self.samplerate), self.samplerate, self.channels) 30. self.audiosave = np.concatenate((self.audiosave, test)) 31. 32. 33. def getAudio(self) -> np.ndarray: 34. return self.audiosave 35. 36. def saveToFile(self, fname : str, bps : Optional[int] = 16) -> None: 37. """ 38. fname - nazwa pliku bez rozszerzenia 39. bps - poziom kwantowania [8, 16, 32] (domyślnie 16) 40. """ 41. 42. if not bps in [8, 16, 32]: 43. raise Exception("bps can be represented only by 8, 16 or 32 bits") 44. 45. data = self.audiosave 46. 47. # poziom kwantyzacji 48. if bps == 8:</pre>					

```
49.         data = np.int8(data/np.max(abs(data)) * np.iinfo("int8").max)
50.     if bps == 16:
51.         data = np.int16(data/np.max(abs(data)) * np.iinfo("int16").max)
52.     if bps == 32:
53.         data = np.int32(data/np.max(abs(data)) * np.iinfo("int32").max)
54.
55.     # zapis do pliku przy użyciu biblioteki scipy
56.     write("{}wav".format(fname), self.samplerate, data)
```

Przetwarzanie C/A

```
1.  def signaltonoise(a, axis=0, ddof=0):
2.      # Zwraca w dB SNR
3.      a = np.asarray(a)
4.      m = a.mean(axis)
5.      sd = a.std(axis=axis, ddof=ddof)
6.      return abs(20*np.log10(abs(np.where(sd == 0, 0, m/sd))))
7.
8.  class Player:
9.
10.
11.     def __init__(self, device=None, samplerate : Optional[int]=48000, channels : Optional[list] = [0,1]) -> None:
12.         """
13.         device : głośnik z biblioteki soundcard (domyślnie, domyślny)\n
14.         samplerate : częstotliwość próbkowania\n
15.         channels : lista kanałów na których ma być odtwarzany dźwięk (dotyczy tylko playData())
16.         """
17.         self.device = device if device != None else sc.default_speaker()
18.
19.         self.samplerate = samplerate
20.         self.channels = channels
21.
22.     def playData(self, data : np.ndarray):
23.         """
24.         data - numpy.ndarray
25.         zwraca -> SNR
26.         """
27.         # odtwarzanie dźwięku
28.         self.device.play(data, self.samplerate, self.channels)
29.         return signaltonoise(data)
30.
31.     def playWav(self, src : str):
32.         """
33.         src : ścieżka do pliku wav
34.         zwraca -> SNR
35.         """
36.
37.         # zapisanie do zmiennej wartości z czytanego pliku
38.         data = read(src)
39.         samplerate = data[0]
40.         # konwersja intów na floaty
41.         data = np.float64(data[1]/np.max(abs(data[1])))
42.         channels = []
43.
44.         for i in range(len(data[0])):
45.             channels.append(i)
46.         # odtwarzanie dźwięku
47.         self.device.play(data, samplerate, channels)
48.
49.         return signaltonoise(data)
```

Podsumowanie wnioski.

Załączony do sprawozdania program spełnia wszystkie założenia zadania. Umożliwia w prosty sposób rejestrację dźwięku oraz jego odsłuch. Zamieszczona jest w nim też możliwość rejestracji do, jak i odczyt dźwięku z pliku. Poprzez przeprowadzenie badań na napisanym przez nas kodzie możemy stwierdzić, że minimalnym akceptowalnym poziomem kwantyzacji jest 16. Przy kwantyzacji na poziomie 8 bitów dźwięk jest zauważalnie gorszej jakości a współczynnik SNR wynosi ~ 1 dB, co według nas jest poniżej akceptowalnej jakości odtwarzanego dźwięku. Dla wartości 16 i 32 jakość dźwięku jest dobra. Przy kwantyzacji na poziomie 32 bitów można zauważyć mniejszą ilość szumów od poziomu 16 bitów co można udowodnić współczynnikiem SNR, który dla mniejszej wartości wskazuje ~ 54 dB a większej ~ 55 dB. Jest to różnica dość mała, ale dla niektórych ludzi zauważalna.