

Telekomunikacja - laboratorium				Studia zaoczne - inzynierskie			
Nazwa zadania		Przetwarzanie analogowo-cyfrowe A/C					
Dzień	poniedziałek		Godzina	14:00		Rok akademicki	2020/2021
Imię i Nazwisko		Kamil Budzyn, 229840					
Imię i Nazwisko		Wiktor Bechciński, 229840					
Opis programu, rozwiązania problemu.							
Celem zadania było zaprojektowanie programu pozwalającego na konwersję dźwięku z postaci analogowej do postaci cyfrowej oraz na odwrót. Program umożliwia nagranie dźwięku do karty dźwiękowej wykorzystując do tego mikrofon a następnie odtworzenie go.							
Najważniejsze elementy kodu programu z opisem.							
MAIN: probkowanie = abs(int(input("Podaj czestotliwosc probkowania [Hz]: "))) //Input czestotliwosci kwantyzacja = int(input("Wybierz poziom kwantyzacji [bit], dostępne opcje to 8, 16, 32: ")) //Input poziomu kwantyzacji print("Wciśnij 'r' aby nagrywać") //Instrukcje dla użytkownika print("Przytrzymaj 'm' aby przestac nagrywać") print("Wciśnij 's' aby zapisac nagranie") print("Wciśnij 'p' aby odtworzyc nagranie") print("Wciśnij 'z' aby odtworzyc zapisane nagranie") REC = Nagrywarka(probkowanie=probkowanie) PLAY = Odtwarzacz(probkowanie=probkowanie) flag = 1 //Ustawienie pierwszej z flag odpowiadającej za wykonywanie programu flag2 = 1 //Ustawienie drugiej z flag odpowiadającej za nagrywanie while flag == 1: if key.is_pressed('r'): //Nagrywanie while flag2 == 1: print('nagrywanie...') if key.is_pressed('m'): print('zakonczono nagrywanie') flag2 = 0 REC.record(1) if key.is_pressed('p'): print('Odtwarzam nagranie...') PLAY.odtworzNagranie(REC.getAudio()) //Odtworzenie nagrania if key.is_pressed('z'): print("Odtwarzam zapisane nagranie...") PLAY.odtworzPlik("NAGRANIE.wav") //Odtworzenie zapisanego nagrania z pliku if key.is_pressed('s'): REC.zapisDoPliku('NAGRANIE', kwantyzacja) //Zapis nagrania do pliku							
C/A: def odtworzNagranie(self, data : np.ndarray): //Odtworzenie zapisanego nagrania self.audio.play(data, self.probkwowanie, self.kanaly) def odtworzPlik(self, src : str): //Odtworzenie nagrania z pliku. data = read(src) probkowanie = data[0] data = np.float64(data[1]/np.max(abs(data[1]))) kanaly = [] for i in range(len(data[0])): kanaly.append(i) self.audio.play(data, probkowanie, kanaly)							
A/C: def record(self, seconds : int): //Nagrywanie if type(self.zapiszAudio) != np.ndarray: self.zapiszAudio = self.audio.record(int(seconds * self.probkwowanie), self.probkwowanie, self.kanaly) else: test = self.audio.record(int(seconds * self.probkwowanie), self.probkwowanie, self.kanaly)							

```
self.zapiszAudio = np.concatenate((self.zapiszAudio, test))

def getAudio(self) -> np.ndarray: //Zapisanie dźwięku bez zapisania do pliku.
    return self.zapiszAudio

def zapisDoPliku(self, plik : str, bps : Optional[int] = 16): //Zapis nagrania do pliku.
    data = self.zapiszAudio
    if bps == 8:
        data = np.int8(data/np.max(abs(data)) * np.iinfo("int8").max)
    if bps == 16:
        data = np.int16(data/np.max(abs(data)) * np.iinfo("int16").max)
    if bps == 32:
        data = np.int32(data/np.max(abs(data)) * np.iinfo("int32").max)
    write("{}.{}.wav".format(plik, self.probkowanie), data)
```

Podsumowanie wnioski.

Program spełnia przedstawione w zadaniu wymagania. Jest w stanie nagrać przez dźwięk z mikrofonu i zapisać go do pliku o rozszerzeniu "WAV", a następnie go odtworzyć.