

Komunikacja pomiędzy komputerami za pomocą dźwięku.

Komunikacja odbywa się poprzez nadawanie dźwięków o ustalonym czasie trwania i częstotliwości dla bitu 0 oraz 1 przez jeden komputer i odbiór komunikatu przez inny komputer.

Wiadomości są zapisywane jak w ramce Ethernetowej, tzn.:

7 x 10101010

1 x 10101011

6 bajtów adresu odbiorcy

6 bajtów adresu nadawcy

2 bajty długości wiadomości

0 – 1500 bajtów wiadomości

4 bajty kodu CRC32

Plik genDzw.py.

Przy uruchamianiu programu podajemy 3 wartości: t – ilość bitów na sekundę, f0 – częstotliwość 0 oraz f1 – częstotliwość 1

Do tworzenia tej ramki z danych podanych na wejściu wykorzystuję funkcjonalności biblioteki bytearray.

Następnie wiadomość ta jest przepuszczana przez 4b5b a na samym końcu przez NRZI. (wszystko oprócz preambuły – pierwszych 8 bajtów)

Do utworzenia dźwięku z wiadomości wykorzystuję bibliotekę Pulseaudio, tworząc obiekt player. Dla danej częstotliwości i czasu nadawania dźwięku tworzę tablicę do której wrzucane są kolejne wartości sinusa tak, abym miał odpowiednią ilość wartości na dany okres sinusa, aby zgadzała się częstotliwość. Otrzymaną wartość mnożę przez 22050 i dodaję do tablicy. I tę tablicę przekazuję do playera, który przetwarza ją na ruchy membrany.

Nasłuchiwanie i rozkodowywanie wiadomości w pliku recv.py:

Przy uruchamianiu programu podajemy 3 wartości: t – ilość bitów na sekundę, f0 – częstotliwość 0 oraz f1 – częstotliwość 1

Tworzony jest obiekt recorder, dostarczany w bibliotece Pulseaudio, który nasłuchuje tak długo aż częstotliwość o maksymalnej mocy jest bliska częstotliwości 0 lub 1 (funkcja recr)

Jeżeli przytrafiła się taka sytuacja, to funkcja getMessage zakłada, że rozpoczęła się preambuła i teraz rozpoczynamy proces szukania najlepszego przesunięcia. Będziemy wykonywali 5 pomiarów z przesunięciem  $1/5 * 1/t$ , tzn słuchamy 1 bitu zapamiętujemy zwrócone częstotliwości oraz „moc” czekamy  $1/5 * 1/t$  czasu i znowu słuchamy 1 bitu zapamiętujemy potrzebne dane i czekamy. Tę procedurę powtarzamy 5 razy. Z tych pięciu przesunięć wybieramy to, którego „moc” była największa, jeżeli częstotliwość dźwięku była bliska 0 albo 1. Jak już mamy wyznaczone najlepsze przesunięcie, to czekamy przesunięcie \*  $1/5 * 1/t$  czasu i powtarzamy tę procedurę 5-krotnie. Potem zakładamy, że już mamy najlepsze przesunięcie i czekamy na koniec preambuły, tzn aż dostaniemy dwie jedynki pod rząd.

Jeżeli tak się stało, to rozpoczynamy słuchanie wiadomości, czyli tak długo aż częstotliwość zwrócona przez recorder jest bliska 0 albo 1.

Następnie jest proces rozkodowywania wiadomości:

1. rozkodowanie z NRZI
2. rozkodowanie z 5b na 4b
3. wyjęcie z kodu bitowego adresu nadawcy, odbiorcy i wiadomości (dodatkowe porównanie, czy otrzymane CRC32 się zgadza, jeżeli nie to nie udało się poprawnie odczytać wiadomości).

Jak działa słuchanie i wyznaczanie najlepszej częstotliwości:

1. recorder słucha i zapisuje odpowiednią ilość próbek do tablicy,
2. na otrzymanej tablicy liczone jest fft
3. szukane jest największy moduł liczby z tej tablicy, i jego indeks jest naszą częstotliwością podzieloną przez czas słuchania.

Aby zapisać lub odebrać wiadomość z pliku .wav należy podać zmienną

\_\_PULSEAUDIO\_WAVFILE\_\_=nazwa.wav