

ASK – LABORATORIUM

SPRAWOZDANIE

ZADANIE 2

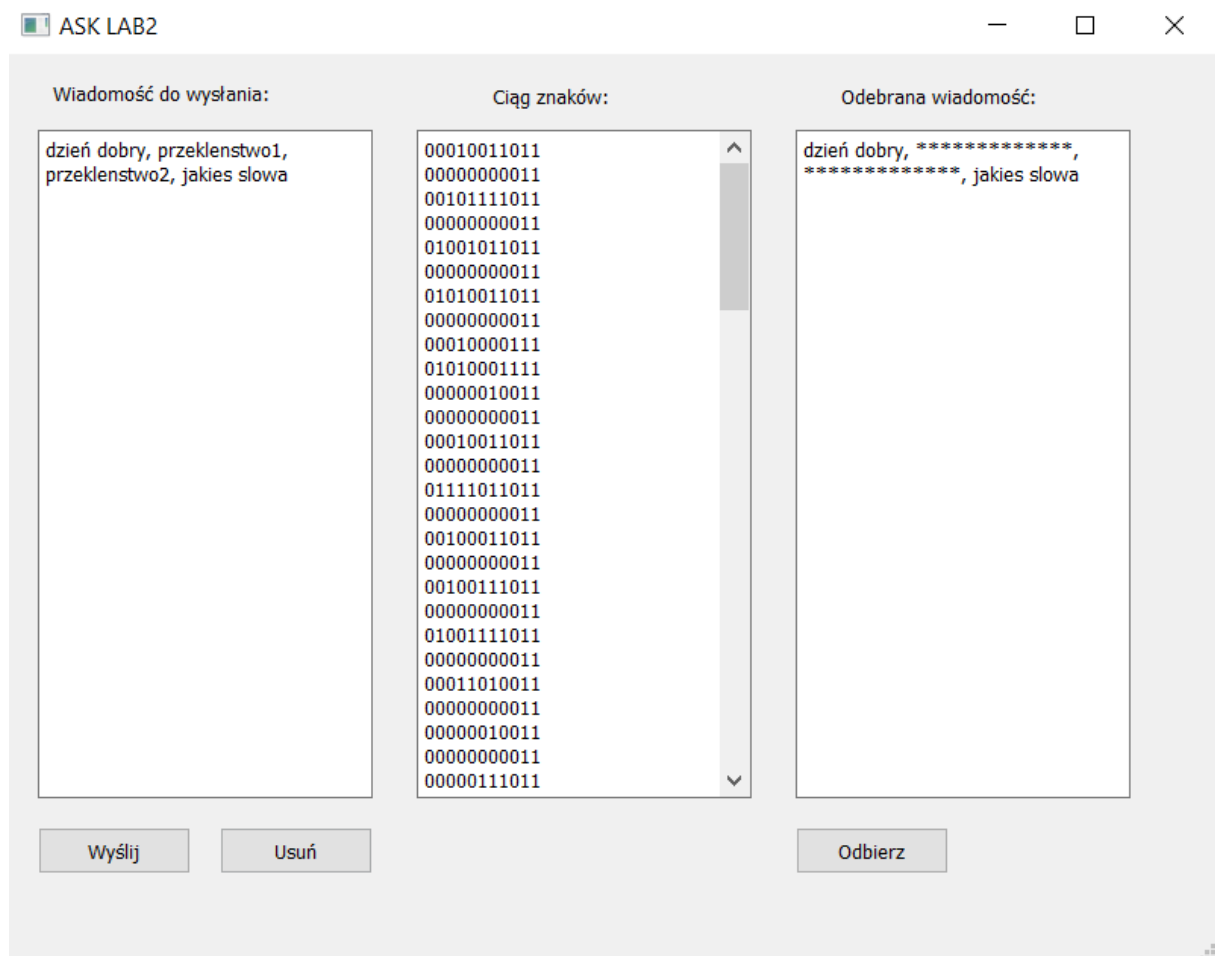
1. Zadanie

Celem zadania drugiego jest wykorzystanie interfejsów dostępnych w komputerze PC do komunikacji pomiędzy dwoma komputerami. Wykorzystując dowolnie wybrany język programowania (wykorzystano język Python) napisać aplikację symulującą transmisję szeregową zgodną ze standardem RS232.

2. Założenia projektowe

- stworzenie okna nadawcy, w którym zostanie wpisana wiadomość do wysłania; wiadomość powinna być ciągiem znaków ASCII
- formatowanie tekstu nadawcy do postaci binarnej: bit startu (0) na początku, następnie bity znaku LSB do MSB, dwa bity stopu (11) na końcu
- używanie dwóch bajtów do przesyłania danych, żeby zachować polskie znaki
- następnie zakodowane dane są przesyłane do okna odbiorcy, do przenoszenia danych użyto zapisywania danych do pliku tekstowego
- przy odebraniu wiadomości w oknie odbiorcy dane są dekodowane (między innymi usuwanie bitów startu oraz stopu), następnie po zamianie na znaki typu ASCII i ocenzeniu odpowiednich słów, wiadomość jest wyświetlana w polu odbiorcy
- cenzurowanie nieodpowiednich słów odbywa się za pomocą porównywania słów zawartych w pliku cenzura.txt ze słowami w odebranej wiadomości
- słowa w pliku cenzura.txt są oddzielone spacjami; ocenzone wiadomości są zamieniane na gwiazdki (*)

3. Działanie programu



Rys.1 Okno programu

Na starcie programu zostaje inicjalizowane okienko z trzema polami tekstowymi oraz trzema przyciskami. Po wpisaniu wiadomości i kliknięciu guzika 'Wyślij' wiadomość typu string jest zakodowywana – każdy znak ASCII staje się 2 ciągami po 11 bitów.

Na początku ciągu znajduje się bit startu (0), bity znaku od najmłodszego do najstarszego oraz bity stopu (11). Jeśli sam znak po binaryzacji posiada więcej niż 8 bitów (np. w przypadku polskich znaków), jego druga część zostaje zakodowana w ten sam sposób w jego drugim ciągu, w przeciwnym razie ciąg wypełniany jest zerami. Tak przygotowany ciąg znaków jest zapisywany do pliku wiadomosc.txt.

Funkcja realizująca kodowanie (użyto tylko wbudowanych bibliotek pythona):

```
def tekst_na_bity(tekst, encoding='utf-8', errors='surrogatepass'):

    bity = bin(int.from_bytes(tekst.encode(encoding, errors), 'big'))[2:]
    bity=bity.zfill(8 * ((len(bity) + 7) // 8))

    bajt1=bity[-8:]
    bajt1=bajt1[::-1]
    bajt1='0'+bajt1+'11'

    if len(bity)>8:
        bajt2=bity[:len(bity)-8]
    else:
        bajt2='00000000'

    bajt2=bajt2[::-1]
    bajt2='0'+bajt2+'11'

    return bajt1,bajt2
```

Rys.2 kodowanie wiadomości

Następnie w celu odebrania wiadomości następuje odwrotna operacja. Po kliknięciu przycisku 'Odbierz', plik wiadomosc.txt zostaje otworzony, wiadomość pobrana, bity startu i stopu zostają usunięte, a 2 bajty składają się na jeden znak typu ASCII.

Przed wyświetleniem wiadomości w oknie odbiorcy, znaki przechodzą przez filtr cenzury. W tym celu otwierany jest plik cenzura.txt, z którego wyjmowane są słowa, które następnie są porównywane z każdym słowem zawartym w odebranej wiadomości. Jeśli nielegalne słowo znajduje się w wiadomości – zostaje zastąpione ciągiem gwiazdek (*) długości zastępowanego słowa w miejscu, w którym się znajdowało. Niecenzuralne słowo zostanie wykryte nawet w połączonych ze sobą słowach.

```

def cenzura(do_cenzury):

    slownik=[]
    with open('cenzura.txt', 'r') as f:
        for line in f.readlines():
            for w in line.split():
                slownik.append(w)

    lista_slow=do_cenzury.split()

    for i in range(len(lista_slow)):
        for j in range(len(slownik)):
            slowo=lista_slow[i]
            znalezione=slowo.find(slownik[j])
            gwiazdki=''*len(slownik[j])
            if znalezione!=-1:
                lista_slow[i]=slowo[:znalezione]+gwiazdki+slowo[(znalezione+le
n(slownik[j])):]
            slowa=''
        for i in range(len(lista_slow)):
            slowa=slowa+lista_slow[i]+' '

    return slowa

```

Rys.3 funkcja cenzurująca nieodpowiednie słowa

4. Wnioski

Aplikacja spełnia założenia przyjęte w punkcie nr 1 sprawozdania. Symulacja działania przesyłu danych przez RS232 została zrealizowana. Do utworzenia okienek aplikacji została użyta biblioteka PyQt5, Qt5 jest standardem dla wielu nowoczesnych aplikacji i posiada wsparcie dla wielu języków programowania.

Aplikacja nie bierze pod uwagę problemów, które mogłyby wystąpić w rzeczywistym przesyśle danych przez RS232.

Kodowanie danych nie jest optymalne – zdecydowanie można by znaleźć sposób, żeby znaki ASCII, które składają się jedynie z 8 bitów nie musiały być konwertowane niepotrzebnie na 2 ciągi po 8 bitów.

Aplikacja jest prosta i przejrzysta, a nielegalne wyrazy zostają wykryte nawet jeśli słowa są ze sobą połączone.

W celu optymalizacji można by używać biblioteki numpy do niektórych operacji na listach, ponieważ biblioteka numpy jest napisana w C.