Telekomunikacja - laboratorium				Studia zaoczne - inzynierskie	
Nazwa zadania	**Protokół Xmodem**				
Dzień	**poniedziałek**	**Godzina**	**14:00-15:30**	**Rok akademicki**	**2020/2021**
Imię i Nazwisko	**Wiktor Bechciński 229840**				
Imię i Nazwisko	**Kamil Budzyn 229850**				

Opis programu, rozwiązania problemu.

Zadane zostało nam przygotowanie programu do przesyłania wiadomosci między portami szeregowymi wykorzystując protokół Xmodem z CRC16. Zadanie polegało na napisaniu funkcji pełniących rolę nadajnika oraz odbiornika wraz z implementacją CRC. Stworzony program należało przetestować przy użyciu innego programu Xmodem.

Najważniejsze elementy kodu programu z opisem.

```
int portSettings(LPCTSTR const_lpctstr) //NADANIE PORTOWI PARAMETROW + OTWORZENIE GO
{
   if (const_handle == INVALID_HANDLE_VALUE)
   {
      return 0;
   }
   else
   {
      const_handle = CreateFile(const_lpctstr, GENERIC_READ | GENERIC_WRITE, 0, NULL, OPEN_EXISTING, 0,
NULL);
      const_dcb.DCBlength = sizeof(const_dcb);
      const_dcb.BaudRate = CBR_9600; // predkosc transmisji
      const_dcb.fParity = TRUE;
      const_dcb.Parity = NOPARITY; // bez bitu parzystosci
      const_dcb.StopBits = ONESTOPBIT; // ustawienie bitu stopu (jeden bit)
      const_dcb.ByteSize = 8; // liczba wysylanych bitów
      const_dcb.fDtrControl = DTR_CONTROL_DISABLE; // Kontrola linii DTR: sygnal nieaktywny
      const_dcb.fRtsControl = RTS_CONTROL_DISABLE; // Kontrola linii RTS: sygnal nieaktywny
      const_dcb.fOutxCtsFlow = FALSE;
      const_dcb.fOutxDsrFlow = FALSE;
      const_dcb.fDsrSensitivity = FALSE;
      const_dcb.fAbortOnError = FALSE;
      const_dcb.fOutX = FALSE;
      const_dcb.fInX = FALSE;
      const_dcb.fErrorChar = FALSE;
      const_dcb.fNull = FALSE;
      return 1;
   }


//Sprawdzanie CRC

int checkCRC(char *point, int count)
{
        int CRCchecksum = 0;

        while (--count >= 0)
        {
                CRCchecksum = CRCchecksum ^ (int)*point++ << 8; // weź znak i dopisz osiem zer
                for (int i = 0; i < 8; ++i)
                {
                        if (CRCchecksum & 0x8000)
                                CRCchecksum = CRCchecksum << 1 ^ 0x1021; // jeśli lewy bit == 1 wykonuj XOR
generatorm 1021
                        else
                                CRCchecksum = CRCchecksum << 1; // jeśli nie to XOR przez 0000, czyli przez to
samo
                }

        }
        return (CRCchecksum & 0xFFFF);
}
}
char checkCRCsign(int CRC, int signNumber) //przeliczanie CRC na postać binarną
{
        int tmp, binary[16];

        for(int i=0;i<16;i++)
                binary[i]=0;
```

```
        for(int i=0;i<16;i++)
        {
                tmp=CRC%2;
                if (tmp==1) CRC=(CRC-1)/2;
                if (tmp==0) CRC=CRC/2;
                binary[15-i]=tmp;
        }

        //obliczamy poszczegolne znaki receivedCRC (1-szy lub 2-gi)
        tmp=0;
        int h;

        if(signNumber==1) h=7;
        if(signNumber==2) h=15;

        for (int i=0;i<8;i++)
                tmp=tmp+binToDec(i)*binary[h-i];

        return (char)tmp;//zwraca 1 lub 2 znak (bo 2 znaki to 2 bajty, czyli 16 bitów)


        // HANDSHAKE
  for(int i = 0; i < 6; i++)
  {
    WriteFile(const_handle, &sign, 1, &sizeOfSign, NULL);
    cout<<"Ustanawianie polaczenia...\n";
    ReadFile(const_handle, &sign, 1, &sizeOfSign, NULL); //sign == SOH || EOT || CAN
    if(sign == SOH)
    {
      cout<<"Polaczenie zostalo ustanowione.\n";
      break;
    }
  }

  if(sign != SOH)
  {
    cout<<"Nie udalo sie ustanowic polaczenia.\n";
    return 0;
  }

  receive.open("receive.txt", ios::out | ios::binary);
  cout<<"Trwa odbieranie pliku...\n";

        //RECEIVE DATA
        while(true)
        {
          if(blockNumber != 0) //zostalo to wczesniej wykonane dla pierwszego bloku jako HANDSHAKE
    {
      ReadFile(const_handle, &sign, 1, &sizeOfSign, NULL); //sign == SOH || EOT || CAN
      if(sign==EOT || sign==CAN) break;
    }

                ReadFile(const_handle, &sign, 1, &sizeOfSign, NULL); //sign == numer bloku
                blockNumber=(int)sign;

                ReadFile(const_handle, &sign, 1, &sizeOfSign, NULL); //sign == dopelnienie do 255
                fillTo255=sign;

                for(int i=0;i<128;i++)
                {
                        ReadFile(const_handle, &sign, 1, &sizeOfSign, NULL); //sign == litera, gdzie jest to
powielone 128 razy czyli jest to blok danych
                        data[i]=sign;
                }
        //SEND DATA
  while(!sendTXT.eof())
  {
                for (int i=0;i<128;i++)
                {
                        bool fillTo128;
                        if(!sendTXT.eof())
       data[i] = sendTXT.get();
       else if(sendTXT.eof())
                        {
                                data[i] = ' ';
                                fillTo128 = true;
```

```
                    }

                    if (fillTo128)
                            for(i;i<128;i++)
                                    data[i] = ' ';

}

while(true)
{
   cout << "Wysylanie " << blockNumber << ". pakietu ";
   WriteFile(const_handle, &SOH, 1, &sizeOfSign, NULL); // Wysylanie znaku początku naglowka.

                    sign=(char)blockNumber;
   WriteFile(const_handle, &sign, 1, &sizeOfSign, NULL); // Wysylanie numeru bloku (1 bajt).

                    sign=(char)255-blockNumber;
   WriteFile(const_handle, &sign, 1, &sizeOfSign, NULL); // Wysylanie dopelnienia (255 - blok).

   for( int i=0; i<128; i++ )
                    {
                            WriteFile(const_handle, &data[i], 1, &sizeOfSign, NULL); // Wysylanie danych
                    }

                    if(gotACK) //obliczanie CRC i wyslanie
   {
      const_ushort=checkCRC(data,128);
      sign=checkCRCsign(const_ushort,1);
      WriteFile(const_handle,&sign, 1, &sizeOfSign, NULL);

      sign=checkCRCsign(const_ushort,2);
      WriteFile(const_handle,&sign, 1, &sizeOfSign, NULL);
   }
```

Podsumowanie wnioski.

Program działa poprawnie, a zaimplementowany protokół Xmodem działa prawidłowo. Sprawdziliśmy to wysyłając oraz odbierając pliki używając programu Hyper Terminal który również posiada zaimplementowany protkół Xmodem. Dzięki zadaniu dowiedzieliśmy się jak działa przesyłanie danych między dwoma komputerami z użyciem protokołu Xmodem.