

Szeregowa transmisja danych

Autor: Grzegorz Żółtek IV FDS

STRESZCZENIE

W pracy na temat „Transmisja szeregowa danych” opisałem takie zagadnienia jak:

- transmisja szeregowa poprzez magistralę RS-232 – jest to podstawowa i najpopularniejsza transmisja danych w sposób szeregowy
- transmisja w systemie Windows – opis sposobu dostępu do portów COM w systemie Windows
- transmisja w systemie Linux – w tym rozdziale opisałem również sposób dostępu do portów szeregowych lecz w przypadku systemów Unixowych
- transmisja poprzez łącze USB – w opisie tego zagadnienia zostały poruszone problemy zalet takiego rozwiązania oraz ewentualnych wad z niego wynikających
- szeregowa transmisja danych w sposób synchroniczny oraz asynchroniczny – opisy tych transmisji dotyczą przede wszystkim sposoby synchronizacji zegarów w przypadku transmisji synchronicznej oraz przesyłania danych w sposób asynchroniczny
- zabezpieczenie danych w transmisji szeregowej – poruszony został problem błędów przy przesyłaniu szeregowym
- szybkość transmisji szeregowej – chodzi tutaj przede wszystkim o taką samą szybkość przesyłu danych przez nadawcę i odbiorcę – by zsynchronizować przesył, oraz szybkość przesyłania danych

SPIS TREŚCI

<u>Streszczenie.....</u>	<u>1</u>
1 <u>Transmisja szeregową.....</u>	<u>3</u>
1.1 <u>Protokoły transmisji szeregowej.....</u>	<u>3</u>
1.2 <u>Transmisja szeregową w systemie Windows.....</u>	<u>5</u>
1.3 <u>Transmisja szeregową RS-23.....</u>	<u>6</u>
1.4 <u>Magistrala szeregową.....</u>	<u>8</u>
1.5 <u>Uniwersalna magistrala szeregową USB.....</u>	<u>8</u>
1.6 <u>Szybkość transmisji szeregowej.....</u>	<u>9</u>
1.7 <u>Szeregową transmisja asynchroniczna.....</u>	<u>9</u>
1.8 <u>Szeregową transmisja synchroniczna.....</u>	<u>10</u>
1.9 <u>Zabezpieczenie danych transmisyjnych w transmisji szeregowej.....</u>	<u>10</u>
1.10 <u>Transmisja szeregową w systemach Unix-owych.....</u>	<u>11</u>
<u>Literatur.....</u>	<u>12</u>

1. Transmisja szeregową

Transmisja szeregową jest to jeden z najtańszych sposobów na przesyłanie danych między dwoma stacjami komputerowymi PC. Jedną z zalet transmisji szeregową jest to, iż nie potrzebuje ona żadnych specjalnych urządzeń do jej realizacji, a oprogramowanie dzięki któremu jest ona realizowana jest dostępne bez żadnych komplikacji.

Stosowane są trzy podstawowe typy transmisji, są to:

SIMPLEX – jest to transmisja realizowana w jednym kierunku

HALFDUPLEX – jest to transmisja realizowana w obu kierunkach ale nie jednocześnie, dane mogą być przesyłane w jednym kierunku a następnie w drugim

FUL-DUPLEX (half-duplex) – jest to transmisja w obu kierunkach. Transmisja duplexowa, może być realizowana poprzez zastosowanie oddzielnej pary przewodów dla każdego z kierunków, bądź poprzez wydzielenie niezależnych kanałów transmisyjnych w ramach tego samego medium. [8]

1.1 Protokoły transmisji szeregową

Jednym z najpopularniejszych protokołów transmisji szeregową jest protokół „XMODEM” Jest to bardzo prosty protokół, który służy do przesyłania plików. Dzięki temu że jest bardzo prosty, oraz jego implementacja jest bardzo łatwa jest on najczęściej stosowanym protokołem do komunikowania się dwóch pecetów. Protokół ten przesyła dane w postaci bloków danych.

Każdy taki blok składa się z :

- Nagłówek
- Danych - 128Bajtów
- Sumy kontrolnej – 1 Bajt

Nagłówek bloku danych pokazuje poniższa tabela:

Tabela 1.

Offset	Długość	Znaczenie
00h	BYTE	Znak początku nagłówka SOH
01h	BYTE	Numer bloku
02h	BYTE	Dopełnienie numeru bloku

Zasada działania protokołu XMODEM

Pierwszy blok który jest transmitowany jest oznaczony numerem 1, po osiągnięciu jednak wartości 0FFh kolejny blok ma numer 0, a dopełnieniem numeru bloku jest numer 255. Suma kontrolna to jednobajtowa suma wszystkich 128 bajtów bez przeniesień. Przed wysłaniem danych komputer oczekuje na sygnał NAK (Negative Acknowledgment) od komputera odbierającego dane, wysyła on ten sygnał co pewien czas, jeśli dane nie są odbierane. Komputer który wysyła dane po otrzymaniu sygnału NAK powinien zacząć wysyłanie pierwszego bloku.

Po każdym odebraniu bloku danych komputer odbierający powinien wysyłać sygnały ACK – jeśli suma kontrolna zgadza się, lub sygnał NAK, gdy się ona nie zgadza. Po wysłaniu całej

wiadomości (wszystkich bloków) nadajnik wysyła sygnał EOT aż do otrzymania potwierdzenia sygnałem ACK.

Kolejnym protokołem służącym do transmisji szeregowej jest protokół YMODEM, jest to unowocześniona wersja protokołu XMODEM o takie elementy jak:

- kontrola błędów CRC-16
- długość bloku 128 – po wysłaniu sygnału SOH lub 1024 bajty – po wysłaniu sygnału STX na początku nagłówka transmitowanego bloku
- po wysłaniu dwóch następujących po sobie sygnałów CAN spowoduje przerwanie transmisji
- przesyłanie kilku plików w jednej sesji
- przesyłanie nazwy pliku

Następnym protokołem jest YMODEM-g.

Jest on wykorzystywany podczas bezpośrednich połączeniach kablowych w celu zwiększenia szybkości transmisji. Protokół YMODEM-g różni się od swojego poprzednika YMODEM takimi szczegółami jak:

- odbiornik nie wysyła sygnału ACK jeśli wykryje błąd, lecz przerywa transmisję dwoma sygnałami CAN
- nadajnik nie oczekuje na potwierdzenie odebrania bloku, lecz po jego wysłaniu wysyła następny

Protokół ZMODEM to kolejna modyfikacja protokołu YMODEM. Zmiany które zostały tam zamieszczone polegają na tym iż można przysyłać ośmiobitowe słowa przez łącza o szerokości siedmiu bitów. Jest to możliwe dzięki specjalnym kodom: [1]

Niektóre kody specjalne wykorzystywane w transmisji szeregowej danych:

Tabela 2

NUUL	00h	Kod pusty
SOH	01h	Początek nagłówka danych
STX	01h	Początek tekstu
ETX	03h	Koniec tekstu
EOT	04h	Koniec transmisji
ENQ	05h	Zapytanie
ACK	06h	Potwierdzenie
BS	08h	Klawisz BACKSPACE
LF	0Ah	Przejdźcie do następnego wiersza
FF	0Ch	Następna strona
CR	0Dh	Powrót kursora
SO	0Eh	Początek zaznaczania
SI	0Fh	Koniec zaznaczania
DLE	10h	Działanie zbliżone do klawisza Esc
DC1	11h	Kod rezerwowowy
NAK	15h	Potwierdzenie negatywne
SYN	16h	Synchronizacja urządzeń
ETB	17h	Koniec bloku danych

SUB	1Ah	Poprawienie błędnie wysłanego znaku
DEL	7Fh	Usunięcie poprzedniego znaku

[1]

1.2 Transmisja szeregową w systemie Windows

W systemie Windows do portów szeregowych COM można realizować tylko poprzez plik. W Windowsie nie występuje bezpośredni dostęp do rejestrów układu odpowiedzialnego za transmisję RS-232. Każda operacja która ma na celu konfigurowanie portów szeregowych COM odbywają się poprzez funkcję Windowsa. System ten jest bowiem odpowiedzialny za nadzorowanie procesów, oraz zadań realizowanych w systemie.

Obsługa portów szeregowych w Windowsie ma wiele zalet, jednak posiada również wady. Zaletami takiego rozwiązania są między innymi:

- bardzo prosta implementacja wszystkich portów transmisyjnych
- możliwość ustawienia swoich buforów wejściowych oraz wyjściowych
- dosyć prosta metoda programowania, oraz szybkie odwoływanie się do funkcji

Wady transmisji szeregowej w Windowsie to przede wszystkim:

- brak jakiegokolwiek możliwości nadzoru nad przebiegiem transmisji w czasie rzeczywistym (w czasie transmisji RS-485 w HALF-DUPLEX)
- różnice przy obsłudze portów COM w zależności od rodzaju Windowsa (inaczej w Windows NT, inaczej w 2000, a jeszcze inaczej w XP, czy 95/98)
- wstrzymywanie transmisji podczas odwoływania się do operacji dyskowych (w Windowsie jest ich bardzo dużo), co powoduje obniżenie szybkości transmisji

Poprzez system operacyjny Windows oraz technologię COM można w łatwy sposób komunikować się. Podczas gdy jedna aplikacja pracuje jako serwer komunikacyjny odpowiadając tym samym za wymianę danych przez port szeregowy COM, odpowiedzi zaś wędrują do aplikacji klient, która nadała zapytanie. Takie rozwiązanie to bardzo duża zaleta, dzięki niej wiele różnych aplikacji (ludzi piszących te aplikacje) może się komunikować poprzez jednego COM-a ze sobą, lub z innymi urządzeniami.

Aby można było przesyłać dane przy pomocy systemu operacyjnego Windows trzeba wykonać szereg operacji programowych takich jak np.

- otwieranie i zamykanie portu
- konfiguracja portu
- obsługa portu

Jak już wcześniej było wspomniane wszystkie te funkcje, typy danych czy struktury, muszą być odpowiednio ustawione. Dla każdej serii Windowsa „operacje” konfiguracji różnią się i to znacznie. [2]

1.3 Transmisja szeregowa RS- 232

Złącze RS-232 jest podstawowym złączem wykorzystywanym w transmisji szeregowej. Znaki składające się z pięciu do ośmiu bitów mogą być przesyłane w sposób szeregowy bit po bicie. Aby przesłać dane przez złącze RS-232 należy zaopatrzyć je w dwa dodatkowe bity sterujące, mianowicie:

- Bit startu , który wyznaczy rozpoczęcie nadawania
- Bit stopu , wyznacza koniec procesu nadawania pojedynczego znaku

Wykorzystywany jest tutaj również bit parzystości, w celu detekcji błędów. Jest to jednak czynność stosowana coraz rzadziej, ponieważ lepszym rozwiązaniem okazuje się być algorytm sum kontrolnych, zapewnia on znacznie większy poziom detekcji błędów w przesyłanym znaku. Bit parzystości kontroluje stosunek zer do jedynek w znaku.

W transmisji szeregowej asynchronicznej stosuje się dwa bity stopu, służy to zwiększeniu odstępu pomiędzy przesyłanymi znakami.

Urządzenia DCE (Data Communication Equipment) są to np. modemy lub też inne urządzenia, które służą w pośredniczeniu w transmisji szeregowej danych. Urządzenia zaś DTE (Data Terminal Equipment) to komputery , lub inne urządzenia które są nadawcami lub odbiorcami informacji.

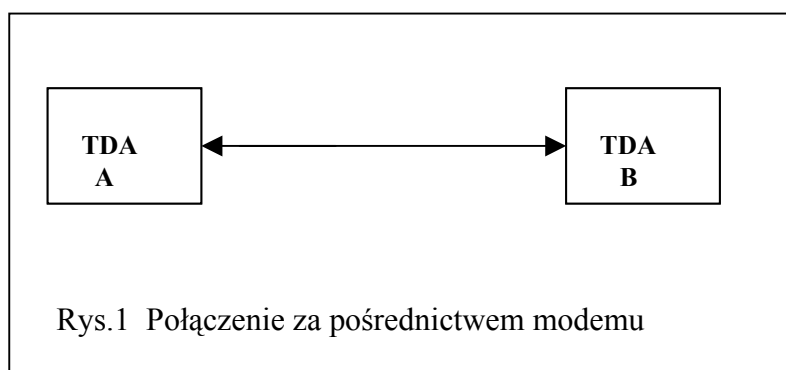
Przesyłanie danych przez magistralę RS-232 zwykle odbywa się na jeden z dwóch sposobów, są to:

- Połączenie przez modem

Używać będziemy tutaj skrótów takich jak (DTE- Data Terminal Equipment; DCE- Data Communication Equipment; DSR- jest to sygnał informujący układ o gotowości do odebrania danych lub ich nadawania; TxD- jest to wyjście nadawanych szeregowych danych; CTS- zezwolenie na nadawanie danych; RTS- sygnał żądający nadawania danych)

Takie rozwiązanie służy do przesyłania danych na większe odległości. Zasada działania urządzeń DTE oraz DCE w przypadku tej transmisji jest następująca:

- podczas nadawania z DTE do DCE urządzenie DTE wymusza stan aktywny na linii DTR(neg), oznacza to iż DTE jest gotowe do współpracy z DCE
- urządzenie DCE potwierdza tę gotowość stanem aktywnym na linii DSR(neg) następnie DTE rozpoczyna komunikację inicjując ją poziomem aktywnym na linii RTS(neg)
- następnym krokiem takiej komunikacji jest odpowiedź modemu (jeżeli jest podłączony do linii) poprzez wygenerowanie poziomu aktywnego na linii CTS(neg)
- teraz już DTE może wysyłać dane po linii TxD
- podczas odbioru (z DCE do DTE) transmisję inicjuje modem
- DCE wymusza teraz stan aktywny na linii DSR(neg)
- Jeżeli jest gotowe DTR do przyjęcia danych to odpowiada poziomem aktywnym na linii DTR(neg)
- Ostatnią czynnością jest przekazanie danych przez DCE przez linię RxD komputera, oczywiście gdy linia DSR(neg) jest w stanie aktywnym



- Połączenie za pomocą kabla

Przy połączeniu za pomocą kabla może być realizowane jedno z kilku połączeń w zależności od złącza które zastosowane jest w komputerze (9 lub 25 pin):

- DB-9 BD-9

RDX(2) ←-----→ TDX(3)
 TDX(3) ←-----→ RDX(2)
 GND(5) ←-----→ GND(5)

- DB-9 DB-25

RDX(2) ←-----→ TDX(2)
 TDX(3) ←-----→ RDX(3)
 GND(5) ←-----→ GND(7)

- DB-25 DB-25

RDX(3) ←-----→ TDX(2)
 TDX(2) ←-----→ RDX(3)
 GND(7) ←-----→ GND(7)

Szeregową transmisję kodów ASCII możemy przeorowadzić na wiele sposobów jednym z nich jest przelączenie prądu o wartości 20mA z częstotliwością odpowiednią dla wybranej szybkości transmisji.

Standard transmisji szeregowej RS-232 definiuje typ złącza oraz przyporządkowanie sygnałów odpowiednim końcówkom co pokazuje poniższa tabela:

Tabela 3

Nazwa sygnału	Wyprowadzenie DB25 (25pin)	Wyprowadzenie DB9 (9pin)	Kierunek DTE←→DCE	Funkcje
TD	2	3	---→	Wysyłanie danych
RD	3	2	←---	Odbieranie danych
RTS	4	7	---→	Gotowość DTE do wysyłania danych
CTS	5	8	←---	Gotowość DCE do odebrania danych
DTR	20	4	---→	Gotowość terminalu
DSR	6	6	←---	Zbiór danych

				gotowy do transmisji
DCD	8	1	←---	Gotowość łącza
RI	22	9	←---	Dzwonek
FG	1			Obudowa
SG	7	5		Masa sygnału

Standard RS-232 został wprowadzony w celu polepszenia i znormalizowania połączeń pomiędzy urządzeniami DTE oraz DCE. Jako zaletę takiego rozwiązania możemy również przyjąć fakt iż do transmisji poprzez łącze RS-232 wystarczy trójprzewodowa magistrala „NULL-MODEM), występuje w złączu DB9 – DB9) [1] ; [3]

1.4 Magistrala szeregową

Na magistralę szeregową interfejsu I2CBUS składają się dwie linie mogące przesyłać dane w obu kierunkach. Jedna z linii jest zarezerwowana dla sygnału danych (Serial Data – SDA) a druga dla sygnału zegarowego (Serial Clock – SCL)

Zdefiniowany został również protokół dla poprawnej wymiany informacji przez tą magistralę.

- zainicjowanie danych może nastąpić tylko wtedy gdy magistrala nie jest zajęta
- w czasie przesyłania danych sygnał na linii SDA nie może się zmieniać , podczas gdy stan sygnału na linii SCL jest wysoki

Określono również odpowiednie warunki przesyłu informacji dla magistrali:

- rozpoczęcie transmisji – zmiana stanu na linii SDA z H na L
- magistrala nie jest zajęta (Bus Not Busy) – linie SDA oraz SCL w stanie H
- zatrzymanie transmisji danych – następuje zmiana na linii SDA z poziomu L na H dokonana podczas wysokiego stanu na linii SCL
- sygnały na linii SDA a raczej ich stany reprezentują tzw. „dane ważne”. Zmiana danych w linii SDA może nastąpić podczas trwania sygnału niskiego na linii SCL. [4]

1.5 Uniwersalna magistrala szeregową - USB

Universal Serial Bus. (USB) - jest to magistrala portów szeregowych które są już dostępne we wszystkich nowszych komputerach

Celem jaki postawili sobie firmy tworzące USB (są to m.in. Intel, Microsoft, IBM i NEC) było skonstruowanie pojedynczego interfejsu, umożliwiłoby komunikację komputera z całym zestawem urządzeń peryferyjnych. Jedną z zalet USB jest to iż można go swobodnie dołączać i odłączać podczas pracy komputera. System operacyjny komputera natychmiast rozpozna nowe urządzenie oraz zainstalować dla niego odpowiednie sterowniki.

Złącze USB wykorzystuje szeregową magistralę danych (serial bus). Wszystkie dane są przesyłane kolejno za pośrednictwem kabla czterozyłowego. Najszybsza dotychczasowa wersja magistrali USB oferuje przepustowość 1,5 MB/s. Można jednak w celu ograniczenia kosztów nabywcy standardu USB opracowany został wolniejszy tryb pracy o przepustowości 200 KB/s. Taka szybkość transmisji jest wystarczająca w przypadku prostych urządzeń przesyłających niewielkie ilości danych.

Niewątpliwie dużą zaletą architektury USB jest możliwość rozbudowy, pozwala ona

bowiem na jednoczesne podłączenie do komputera nawet 127 urządzeń peryferyjnych. Odbywa się to za pośrednictwem tzw. **hubów**.

Takie rozwiązanie daje nam ograniczenie liczby kabli na tylnym panelu komputera, co niewątpliwie jest zaletą.

Podłączenie urządzenia USB do komputera możemy realizować w sposób kaskadowy. [4]

1.6 Szybkość transmisji szeregowej

Nawet w transmisji asynchronicznej urządzenia nadawcze i odbiorcze muszą być w jakiś sposób „zsynchronizowane” – muszą więc pracować z „jednakową szybkością” (to znaczy wiedzieć ile „trwa” transmisja pojedynczego bitu).

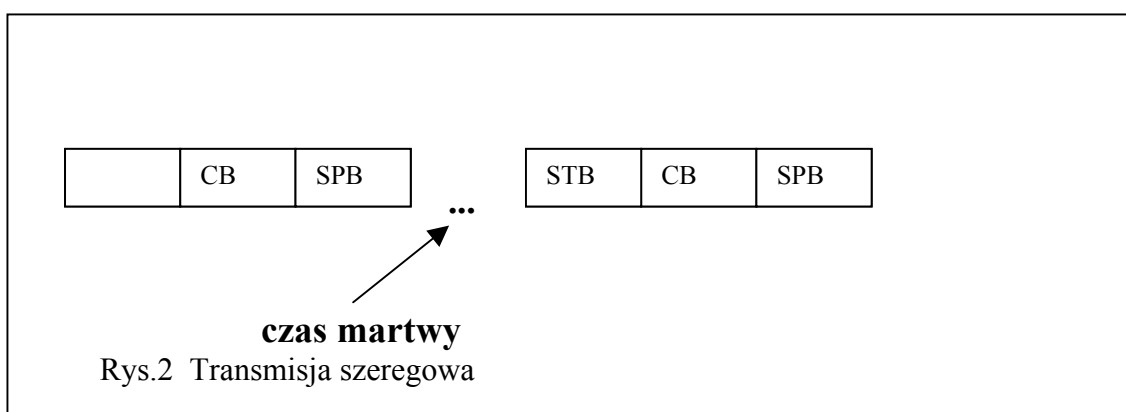
Szybkość transmisji podawana jest w bitach na sekundę - **bps** (ang. *bits per second*),.

Najczęściej używane szybkości w transmisji szeregowej mieszczą się w następującym szeregu: 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 oraz 115200 bps.

Urządzenia nadawcze i odbiorcze muszą zatem pracować z jednakową szybkością. Zazwyczaj jest ona taka sama w obu kierunkach.[5]

1.7 Szeregowa transmisja asynchroniczna

Zasada działania asynchronicznej transmisji szeregowej



Transmisja rozpoczyna się od przesłania bitu startu (bitSTB) następnie przesyłany jest znak (oznaczony CB) – zwykle 7 lub 8 bitów danych, transmisję kończy bit stopu(STB). Po czasie martwym procedura jest powtarzana.

Na początku transmisji przesyłany jest jednostkowy impuls (zero lub jeden), musi on być przeciwny do sygnału zastosowanego w czasie martwym. Odbiorca takiego bitu startowego wie że w takim odcinku czasu będą nadawane następne bity tworzące przesyłaną informację następuje więc krótkie zsynchronizowanie nadawcy z odbiorcy. Bity stopu w zależności od rodzaju zastosowanego rozwiązania trwają od jednej do kilku jednostek czasu.

Do podstawowych wad transmisji asynchronicznej należą:

- istnienie czasu martwego, które ogranicza wydajność przesyłu nawet do 60%
- szybkość transmisji jest ograniczona do kilkudziesięciu Kb/s

Zaletami takiej transmisji są:

- niezależność zegarów nadawcy i odbiorcy
- zamknięcie transmisji w ścisłych ramach czasowych

Transmisja nazywana jest **asynchroniczną** gdyż zakłada się, że dane mogą pojawiać się w dowolnej chwili i będą natychmiast transmitowane do odbiorcy.

Z tego też powodu rozpoczęcie transmisji danych musi być zasygnalizowane w jakiś sposób.

W tym celu używamy dodatkowych dwóch bitów:

- bit startu – 0
- bit stopu - 1.

Znak A wysyłany w sposób szeregowy „wygląda” następująco:

```

1
0  SP 1  00000  1T

```

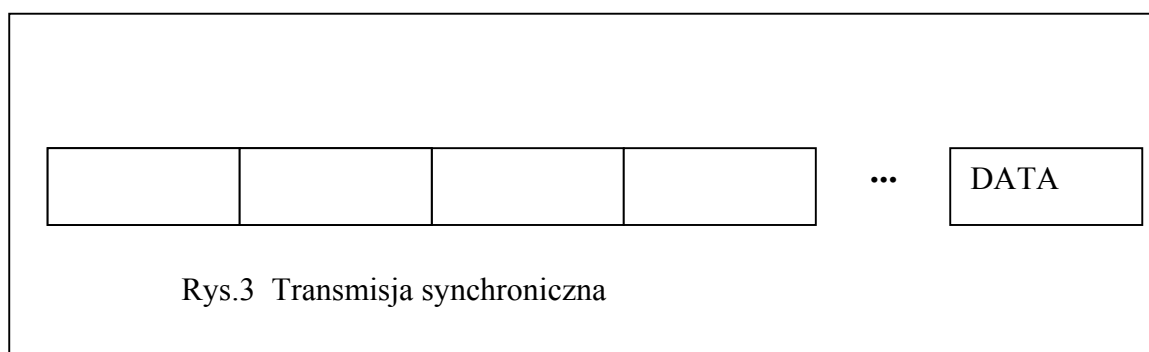
Oznaczenia użyte w powyższym wykresie to:

- S** - bit startu,
- P** - bit parzystości,
- T** - bit stopu.

[6] ; [8]

1.8 Szeregową transmisja synchroniczna

Zasada działania transmisji synchronicznej



Rys.3 Transmisja synchroniczna

W transmisji synchronicznej specjalna preambuła dokonuje zsynchronizowania nadawcę i odbiorcę. Preambuła ta to ciąg znaków zerojedynkowych o ściśle określonym czasie trwania i ilości. Początek nadawania to preambuła (zazwyczaj jest to nie więcej niż 25 impulsów) – SYNC. Po synchronizacji następuje przesłanie danych (DATA). Preambuła synchronizująca jest powtarzana tylko wtedy gdy następuje rozsynchronizowanie – co może się objawiać wzrostem błędów w transmisji.

Zalety transmisji synchronicznej to:

- brak przerw w transmisji
- nielimitowana szybkość transmisji

Wadami takiej transmisji są:

- konieczność zsynchronizowania zegarów nadawcy i odbiorcy
- urządzenia nadawczo-odbiorcze muszą być wyposażone w bufory [8]

1.9 Zabezpieczenie danych transmisyjnych w transmisji szeregowej

Przesyłanie danych przez łącze RS-232 prawie zawsze odbywa się bez błędów komunikacyjnych, niebezpieczeństwo ich wystąpienia pojawia się na łączach komutowanych

i radio- modemach. Potrzebne jest więc zabezpieczenie przed skutkami takich błędów. Obecnie istnieje dwa sposoby zabezpieczeń:

- TimeOut - kontrola czasowa (czas przesyłania ramki jest ściśle określony)
- Kontrola całej ramki jednym z kilku algorytmów sumy kontrolnej

Możemy wyróżnić tutaj trzy najbardziej znane algorytmy obliczania sumy kontrolnej:

1. algorytm wielomianowy CRC-16 który wygląda następująco: $x^{16}+x^{15}+x^2+1$
2. algorytm wielomianowy CRC-CCITT o postaci: $x^{16}+x^{12}+x^5+1$
3. arytmetyczna suma wszystkich przesłanych bitów

Skuteczność pierwszych dwóch algorytmów wielomianowych jest prawie idealna, zapewniają bowiem one poziom ufności przesyłanych danych na granicy 99,99%.

1.10 Transmisja szeregowa systemach Unix-owych

Wszystkie systemy Unix-owe , a więc także i Linux różnią się od systemu operacyjnego Windows budową ioraz sposobem pracy w systemie.

Zarówno Windows, jak i każda odmiana Unix-a mają wspólną cechę, jest nią obsługa portów szeregowych. Tak jak w Windowsie również i tutaj nie występuje bezpośrednie odwoływanie się do rejestrów kości, które są odpowiedzialne za transmisję szeregową. Dostęp do portów szeregowych jest tylko poprzez funkcję które odwołują się do jądra systemu (KERNEL-a). Jednak już nazwy portów różnią się w zależności którego systemu używamy , co obrazuje poniższa tabela:

Tabela 4

	DOS	Windows	Unix	Linux
Nazwa portu	COM1	COM1	dev	ttyS0
	COM2	COM2	dev	ttyS1
	COM3	COM3	Dev	ttyS2

Proces ustawienia portu do transmisji szeregowej sprowadza się do przekazania odpowiednio ustawionych zmiennych przez funkcję do jądra systemu.

Podczas operacji zamykania i otwierania portów należy użyć nazwy dev lub ttyS(0,1,2), w zależności od tego czy nasz system operacyjny to Unix czy Linux.

Pierwszą z czynności jakie należy podjąć podczas szeregowego przesyłania danych jest połączenie komputerów najprostszym z przewodów jakim jest NULL MODEM. Możemy wtedy zauważyć czy nie występują zwłoki czasowe w przekazywaniu danych. Zarówno w systemie Linux jak i Unix obsługa portu jest realizowana poprzez POSIX – termios interface. Systemy Unix-owe posiadają wiele parametrów służących do transmisji szeregowej danych.[7]

Literatura

- [1] Sebastian Majewski – Artykuł „Dla praktyków” w czasopiśmie komputerowym „Pckurier” nr 22/1997.
- [2] <http://republika.pl/serialtransmit/windows.html>
- [3] http://republika.pl/serialtransmit/rs_232.html
- [4] http://www.immt.pwr.wroc.pl/export_hp/tool/node126.html
- [5] http://www.immt.pwr.wroc.pl/export_hp/tool/node127.html
- [6] http://republika.pl/tylek15/sieci_pliki/node126.html
- [7] <http://republika.pl/serialtransmit/linux.html>
- [8] materiały z wykładu „Sieci komputerowe”