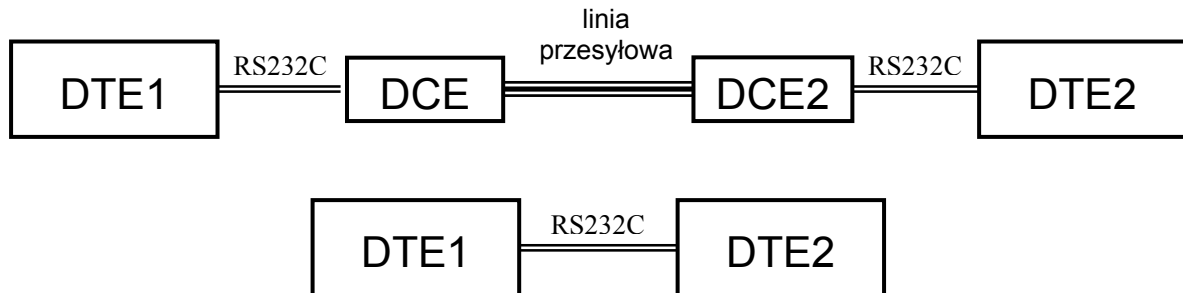


# System interfejsu RS – 232C

opracowali P. Targowski i M. Rębarz

Standard RS – 232C (*Recommended Standard*) został ustanowiony w 1969 r. przez Electronic Industries Association.

Definiuje on sposób nawiązania i przeprowadzenia łączności między dwoma urządzeniami tzw. DTE (*Data Terminal Equipment*) za pośrednictwem modemów tzw. DCE (*Data Communication Equipment*) lub bez nich. Jest to obecnie standard łącza szeregowego do połączenia komputera PC z urządzeniami zewnętrznymi.



Połączenia między urządzeniami (DTE – DCE lub DTE – DTE) dokonuje się za pomocą złączy 25 – stykowych lub 9 – stykowych.

Najważniejsze linie magistrali RS– 232C podane są w tabeli.

Numer przewodu i styku				Opis sygnału		Kierunek
Obwód	RS-232	Styk DB25	Styk DB9	Nazwa polska	Nazwa angielska	DTE - DCE
101	PG	1	-	Masa ochronna	Protective Ground	-
102	SG	7	5	Masa sygnałowa	Signal Ground	-
103	TxD	2	3	Dane nadawane	Transmitted Data	→
104	RxD	3	2	Dane odbierane	Received Data	←
105	RTS	4	7	Żądanie nadawania	Request to Send	→
106	CTS	5	8	Gotowość do nadawania	Clear to Send	←
107	DSR	6	6	Gotowość DCE	Data Set Ready	←
108	DTR	20	4	Gotowość DTE	Data Terminal Ready	→
109	DCD	8	1	Poziom sygnału	Carrier Detected	←

## Opis linii

### **Linie Danych (obowiązuje logika negatywna):**

**TxD** – dane nadawane.

**RxD** – dane odbierane

### **Linie sterujące (obowiązuje logika pozytywna):**

**RTS** – żądanie nadawania danych zgłaszane przez terminal DTE

**CTS** – gotowość do nadawania zgłaszana przez modem DCE (przesyła potwierdzenie odebrania sygnału RTS)

**DSR** – gotowość modemu DCE do dalszej współpracy z DTE (aktywny przez cały czas trwania połączenia)

**DTR** – gotowość DTE do dalszej współpracy z DCE (aktywny przez cały czas trwania połączenia)

**DCD** – sygnał wykrycia przez modem fali nośnej (oznacza, że łączy się on z innym modemem)

### **Linie masy:**

**SG** – masa sygnałowa

**PG** – masa ochronna połączona z obudową urządzenia

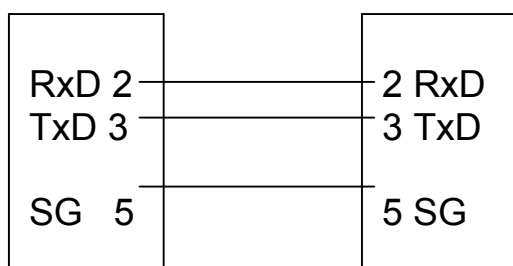
Istnieje wiele sposobów wykorzystania różnych podzbiorów tych linii. W najprostszym przypadku wykorzystuje się tylko 3 linie:

**TxD** – dane nadawane

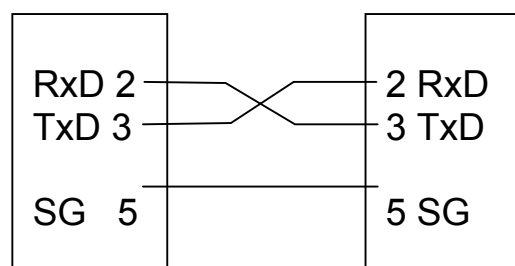
**RxD** – dane odbierane

**SG** – masa sygnałowa

Jeżeli łączymy urządzenie typu DTE (komputer) z urządzeniem z DCE (modem) to styki łączymy wprost. Jeżeli komputer do komputera to używamy tzw. konfiguracji *null modem*. Na przykład dla wtyków 9-cio pinowych:



DTE – DCE



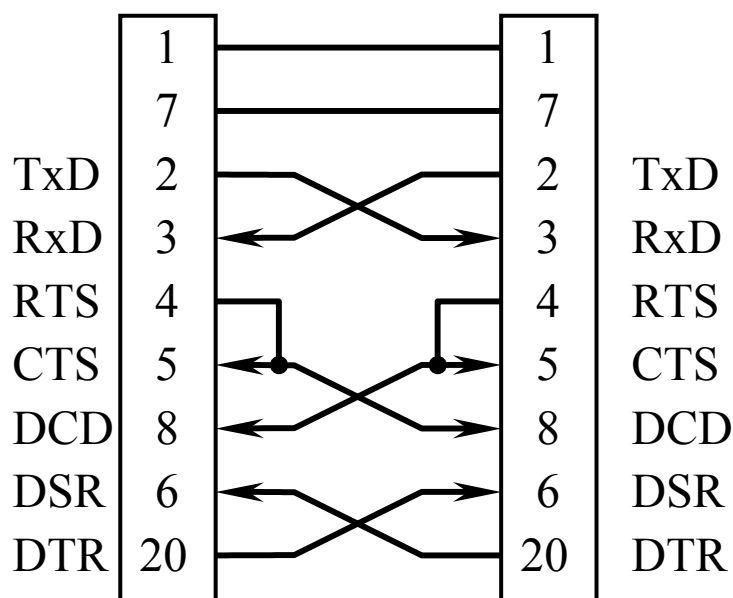
DTE – DTE (*null modem*)

Linie DTR, DSR, RTS, CTS i DCD umożliwiają synchronizację transmisji tak, dane były wysyłane zawsze wtedy, gdy urządzenie po drugiej stronie połączenia jest gotowe do ich odbioru. Prosty przykład transmisji jednokierunkowej (simplex) może wyglądać następująco:

1. Aby transmisja była możliwa oba urządzenia muszą ustawić linie gotowości na „1”. Gotowość komputera (DTR) i modemu (DSR) musi być utrzymywana na „1” przez całą transmisję.
2. Komputer ustawia RTS (żądanie nadawania) na „1”
3. W odpowiedzi na RTS modem ustawia linię CTS (gotowość do nadawania) na „1”
4. Widząc to komputer rozpoczyna nadawanie

symbol	obwód	stan linii	uwagi
DTR	108/2		komputer gotów
DSR	107		modem gotów
RTS	105		żądanie nadawania
CTS	106		gotowość do nadawania
TxD	103		transmisja danych do modemu

Bardziej rozbudowany kabel modemu zerowego (dla wtyku 25cio pinowego ) umożliwiający **transmisję dwukierunkową (duplex)** jest pokazany poniżej

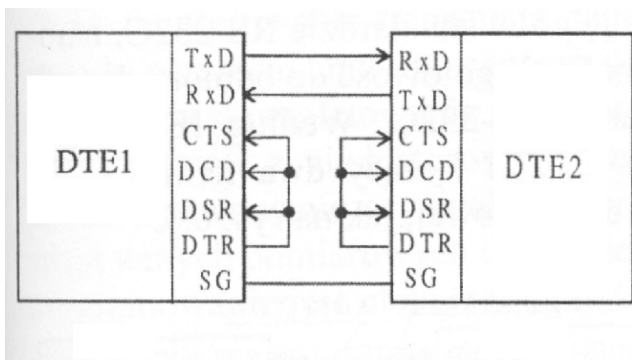
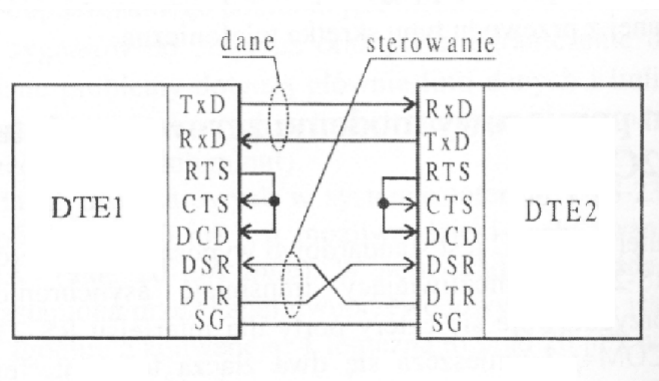
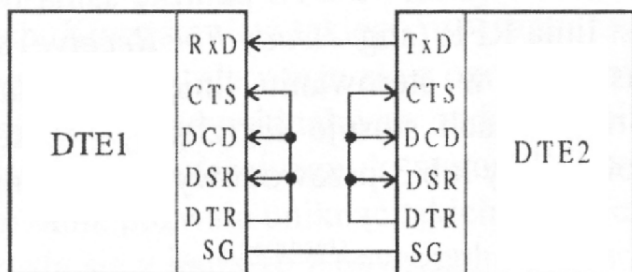


W tym przykładzie oba komputery wystawiają sygnał gotowości DTR, który dzięki połączeniu „na krzyż” widziany jest przez drugi komputer jako sygnał DSR. Dzięki połączeniu RTS do swojego CTS, automatycznie po ustawieniu RTS na „1” komputer widzi „odpowiedź” CTS=1 i rozpoczyna wysyłanie danych, o ile linia DCD (kontrolowana przez drugi komputer) jest w stanie „1”. W ten sposób ustawienie „0” na RTS wstrzymuje przepływ danych.

#### Rodzaje transmisji ze względu na kierunek przepływu danych:

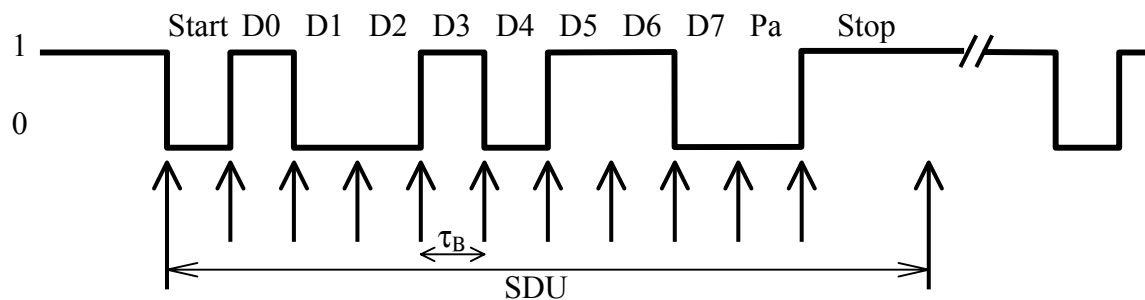
<b>Transmisja simpleksowa</b>	jednokierunkowa transmisja, w której tylko DTE1 przekazuje dane DTE2 lub tylko DTE2 przekazuje dane DTE1
<b>Transmisja półdupleksowa</b>	dwukierunkowa transmisja niejednoczesna np. najpierw DTE1 – DTE2, a następnie DTE2 – DTE1
<b>Transmisja duplexowa</b>	dwukierunkowa transmisja jednoczesna, w tym samym czasie DTE1 – DTE2 i DTE2 – DTE1

## Inne przykłady stosowanych połączeń kabla RS-232



## Transmisja liniami danych

Przesyłanie informacji następuje w sposób szeregowy bit po bicie. Stany logiczne 0 i 1 kodowane są stanami napięć (lub wartościami prądu). Najczęściej przesyłane są znaki danych zapisane w kodzie ASCII. Każdy znak danych zawiera od 5 do 8 bitów i poprzedzony jest bitem START, a zakończony bitem kontroli parzystości (Pa) i 1 do 2 bitami STOP. Bity danych wraz z bitem kontrolnym i bitami synchronizacji (start, stop) tworzą tzw. jednostkę informacyjną SDU (*Serial Data Unit*).



Bity danych są przesyłane w kolejności od najmniej znaczącego D0, do najbardziej znaczącego. Opcjonalny bit parzystości ma wartość logiczną równą sumie modulo 2 wszystkich bitów danych. W przykładzie powyżej zastosowano następujące parametry transmisji:

- 8 bitów danych
- bit parzystości
- 2 bity stopu

i nadano liczbę:  $01101001_b = 69_h = 105_{dec} = „i”$ . Liczba zawiera parzystą ilość jedynek i dlatego bit parzystości Pa = 0.

Bit START uruchamia zegar zapewniający właściwą synchronizację odczytu. Częstota pracy tego zegara ( $=1/\tau_B$ ) określa szybkość transmisji. Typowe prędkości transmisji wynoszą: 300,600,1200, 2400, 4800, 9600, 19 200 b/s (bitów/sek).

Do sterowania przebiegiem transmisji stosuje się często protokół transmisji **XON – XOFF**. Wykorzystuje on dwa znaki (XON = 11hex i XOFF = 13hex) sterujące z zestawu ASCII przekazywane linią danych TxD – RxD. Odbiornik danych sygnalizuje za pomocą znaku XON gotowość przyjęcia dalszych znaków natomiast pojawienie się znaku XOFF wstrzymuje transmisją danych.

### Parametry elektryczne sygnałów

Napięcia na liniach danych:

od -15V do -3V : **1** logiczna

od 3V do 15 V: **0** logiczne

Napięcia na liniach sterujących:

od -15V do -3V : **0** logiczne

od 3V do 15 V: **1** logiczna

Czas przejścia przez obszar przejściowy  $< 3\%$  czasu trwania bitu. Szybkość zmian sygnałów w liniach  $> 30 \text{ V}/\mu\text{s}$ .

## Bezpośrednie programowanie UART

Programując wymianę informacji poprzez port szeregowy można korzystać z gotowych procedur, albo bezpośrednio sterować odpowiednim portem poprzez adresowanie wejścia/wyjścia. Adresy bazowe dla standardowych portów szeregowych są następujące:

**COM1 : 03F8**

**COM2 : 02F8**

**COM3 : 03E8**

**COM4 : 02E8**

Pod tym adresem wysyła się (wtedy nazywa się **Transmitter Holding Register TSR**) i odbiera się dane (wtedy nazywa się **Receiver Data Register RDR**) jeżeli LCR:b7=0 albo ustawia szybkość transmisji (LCR:B7=1) – wtedy nazywa się **Baud Rate Divisor BRDL**.

Adres bazowy +1 jeżeli (LCR:B7=1) również ustawia szybkość transmisji: BRDH:

Baud Rate	bajt w BRDH (hex)	bajt w BRDL (hex)
50	09	0
110	04	17
300	01	80
1200	0	60
2400	0	30
4800	0	18
9600	0	0C
19200	0	06

Adres bazowy +1 dla LCR:b7=0 (**IER**) oraz adres bazowy+2 (**IIE**) służą do kontroli ewentualnych przerw. Aby wyłączyć generowanie przerw wpisać 0 do IER.

Adres bazowy + 3 (**line control register LCR**) służy do kontroli sposobu transmisji danych:

b7	b6	b5	b4	b3	b2	b1	b0
alternate	break	kontrola parzystości			stop	ile bitów?	
norm: 0	disable: 0	ignore: 000			1 bit: 0	5 bitów: 00	
	enable: 1	odd: 100			1.5 bits: 1	6 bitów: 01	
BRDL/ BRDH: 1		even 110			2 bits : 1	7 bitów: 10	
		mark: 101				8 bitów: 11	
		space 111					

Adres bazowy + 4 (**modem control register MCR**) - rejestr kontroli modemu

b7	b6	b5	b4	b3	b2	b1	b0
x	x	x	UART loopback	user #2	user #1	RTS active	DTR active

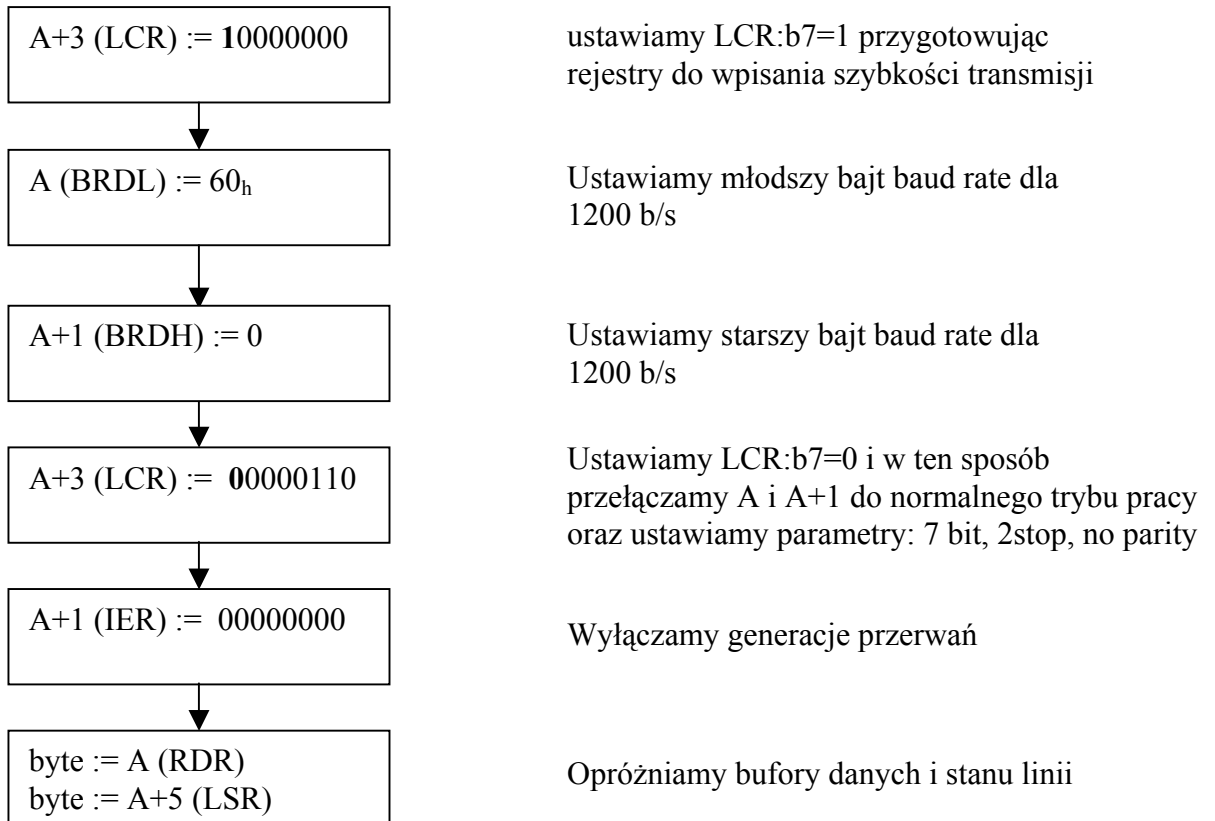
Adres bazowy + 5 (**line status register LSR**) – rejestr kontroli linii

b7	b6	b5	b4	b3	b2	b1	b0
time out	TSR empty	THR empty OK. to output	break detect	framing error	parity error	overrun error	data received - byte in RDR

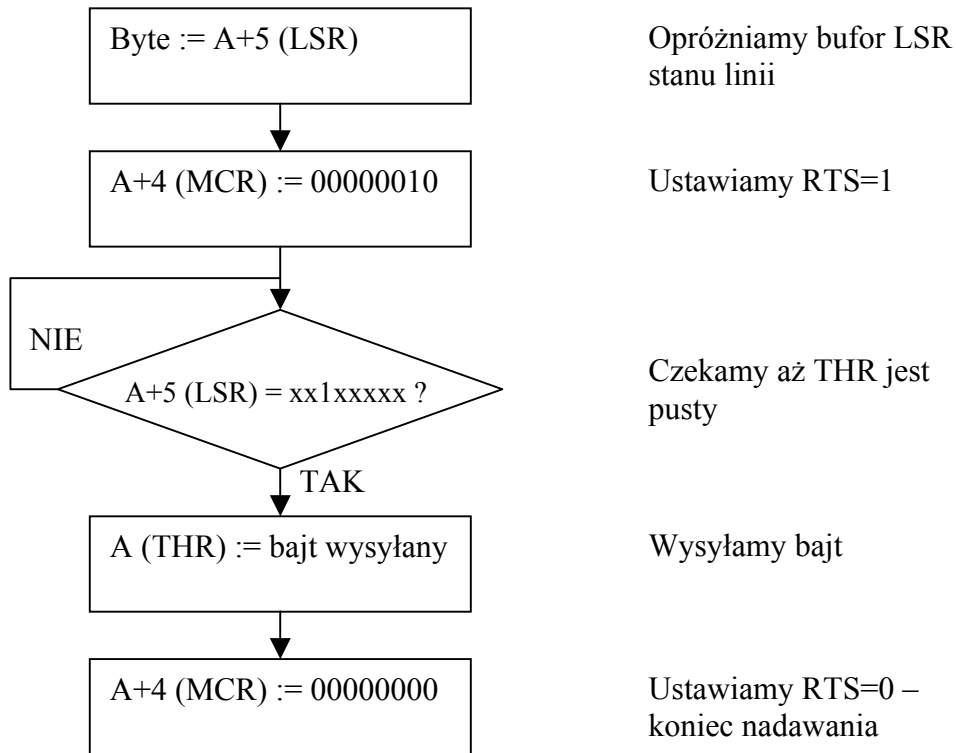
Adres bazowy + 6 ( **modem status register MSR** )

7	b6	b5	b4	b3	b2	b1	b0
DCD set high	RI set high	DSR set high	CTS set high	change in DCD	change in RI	change in DSR	change in CTS

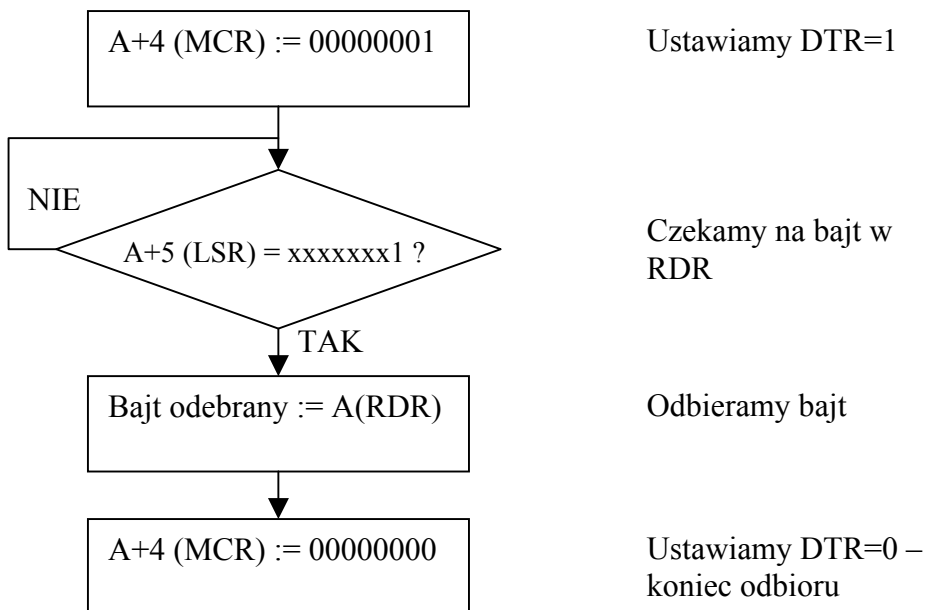
**Przykład inicjalizacji portu:**



### Przykład nadawania 1 bajtu:



### Przykład odbioru 1 bajtu





**Wybrana literatura:**

1. W. Nawrocki „Komputerowe systemy pomiarowe” WKŁ Warszawa 2002
2. W. Nowakowski, red. „Systemy interfejsu w miernictwie” WKŁ Warszawa 1987
3. W. Tłaczała „Środowisko LabVIEW<sup>TM</sup> w eksperymencie wspomaganym komputerowo” WNT Warszawa 2002
4. T.R. Dettmann „DOS Programmer’s Reference”, Que 1988