helion.pl
KSIĘGARNIA INTERNETOWA · WYDAWNICTWO

IDŹ DO
PRZYKŁADOWY ROZDZIAŁ
SPIS TREŚCI

KATALOG KSIĄŻEK
KATALOG ONLINE
ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK
DODAJ DO KOSZYKA

CENNIK I INFORMACJE
ZAMÓW INFORMACJE O NOWOŚCIACH
ZAMÓW CENNIK

CZYTELNIA
FRAGMENTY KSIĄŻEK ONLINE

Interfejsy sprzętowe komputerów PC



Autor: 83-7361-663-2
Tłumaczenie: Radosław Meryk (wprowadzenie, rozdz. 1 – 6), Przemysław Szeremiota (rozdz. 7 – 10)
ISBN: MichaelGook
Tytuł oryginału: PC Hardware Interfaces: A Developers Reference
Format: B5, stron: 536

Dowiedz się, w jaki sposób komputery PC komunikują się z innymi urządzeniami

- Rodzaje i zadania interfejsów
- Magistrale i karty rozszerzeń
- Dedykowane interfejsy urządzeń peryferyjnych

Komputer PC może współpracować z różnymi urządzeniami – od drukarek, modemów i innych komputerów, aż do urządzeń pomiarowych, siłowników i innych sterowanych z poziomu komputera. Połączenie komputera z urządzeniem zewnętrznym jest realizowane za pośrednictwem interfejsów – złączy umożliwiających komunikację dwóch urządzeń.

Książka „Interfejsy sprzętowe komputerów PC" to kompletne omówienie wszystkich interfejsów stosowanych do łączenia komputerów PC z urządzeniami peryferyjnymi. Przedstawia zarówno rodzaje interfejsów, jak i zasady ich działania, typy przesyłanych przez nie sygnałów i reguły przydzielania im pamięci w komputerze. Omawia mechanizmy transmisji przez każdy z zaprezentowanych interfejsów oraz sygnały sterujące działaniem ich kontrolerów.

- Zasada działania interfejsów
- Interfejsy COM i LPT
- Złącza bezprzewodowe – IrDA i Bluetooth
- Magistrale USB, FireWire i SCSI
- Złącza kart rozszerzeń
- Interfejsy klawiatury i myszy
- Złącza pamięci masowych i sieciowe

Jeśli chcesz stworzyć aplikację, która będzie wykorzystywać jeden z interfejsów komputera PC, lektura tej książki będzie strzałem w dziesiątkę.

Wydawnictwo Helion
ul. Chopina 6
44-100 Gliwice
tel. (32)230-98-63
e-mail: helion@helion.pl

Spis treści

Wprowadzenie w tematykę interfejsów 7
Ogólna struktura komputerów 7
Urządzenia peryferyjne 7
Hierarchia połączeń 8
Organizacja interfejsów 10
Typy przesyłanych informacji 10
Interfejsy równoległe i szeregowe 11
Sygnały i media transmisji 16
Galwaniczna separacja podłączonych urządzeń 18
Interakcje pomiędzy urządzeniami i topologia urządzeń 19
Kontrola poprawności transmisji i jej niezawodność oraz sterowanie przepływem 22
Parametry czasowe i synchronizacja 24
Ewolucja sposobów podłączania urządzeń peryferyjnych 25
Interfejsy poziomu systemu 25
Interfejsy urządzeń peryferyjnych 31
Wybór interfejsu 33

Rozdział 1. Interfejs równoległy: port LPT 35
1.1. Tradycyjny port LPT 36
1.2. Ulepszenia portu równoległego 40
1.3. Standard IEEE 1284 40
1.3.1. Tryb półbajtowy 41
1.3.2. Bajtowy tryb wejściowy 43
1.3.3. Tryb EPP 44
1.3.4. Tryb ECP 49
1.3.5. Tryby i rejestry ECP 52
1.3.6. Negocjacja w standardzie IEEE 1284 56
1.3.7. Interfejsy fizyczne i elektryczne 57
1.3.8. Rozwój standardu IEEE 1284 60
1.4. Obsługa systemowa portów LPT 61
1.5. Funkcje portu równoległego oraz obsługa Plug-and-Play 62
1.6. Zastosowania portu LPT 62
1.7. Konfiguracja portów LPT 64
1.8. Rozwiązywanie problemów z działaniem portu równoległego 66

Rozdział 2. Interfejs szeregowy: port COM 69
2.1. Interfejs RS-232C 70
2.2. Interfejsy pomocnicze i konwertery poziomów 75
2.3. Tryb asynchroniczny 80
2.4. Sterowanie przepływem danych 81
2.5. Układy scalone asynchronicznych nadajniko-odbiorników (UART) 83
2.6. Obsługa portów COM w systemie operacyjnym 90
2.7. Konfiguracja portów COM 91
2.8. Zastosowania portów COM 91
2.9. Porty COM i PnP 93
2.10. Problemy z działaniem portów COM i ich rozwiązywanie 94
2.10.1. Testowanie konfiguracji 94
2.10.2. Testowanie działania 95
2.10.3. Brak zasilania interfejsu, czyli dlaczego nie działa mysz 97

Rozdział 3. Interfejsy bezprzewodowe 99
3.1. Port podczerwieni — IrDA 99
3.2. Radiowy interfejs Bluetooth 103

Rozdział 4. Magistrale szeregowe: USB i FireWire 109
4.1. Uniwersalna magistrala szeregowa: USB 110
4.1.1. Architektura USB 110
4.1.2. Model transmisji danych 116
4.1.3. Transakcje magistrali 119
4.1.4. Przepustowość magistrali i urządzenia 127
4.1.5. Synchronizacja podczas transmisji izochronicznych 133
4.1.6. Interfejs fizyczny 137
4.1.7. Urządzenia USB 146
4.1.8. Koncentratory USB 148
4.1.9. Host 159
4.1.10. Konfiguracja i zarządzanie urządzeniami 171
4.1.11. Zastosowania interfejsu USB 178
4.2. Magistrala IEEE 1394: FireWire 194
4.2.1. Fizyczna warstwa sieci 195
4.2.2. Protokół IEEE 1394 196
4.2.3. Urządzenia i adaptery interfejsu 1394 199
4.2.4. Zastosowania interfejsu 1394 200

Rozdział 5. Magistrala SCSI 203
5.1. Równoległe interfejsy SCSI 206
5.1.1. Protokół magistrali 207
5.1.2. Sterowanie interfejsem i wykonywanie poleceń 213
5.1.3. Kable, złącza i sygnały 216
5.1.4. Terminatory 224
5.1.5. Konfiguracja urządzeń SCSI 226
5.1.6. Podłączanie urządzeń do magistrali SCSI 228
5.2. Interfejs Fibre Channel 233
5.3. Adapter hosta SCSI 234

Rozdział 6. Magistrale i karty rozszerzeń wejścia-wyjścia 237
6.1. Magistrale PCI oraz PCI-X 238
6.1.1. Wyliczanie urządzeń 241
6.1.2. Protokół magistrali 243
6.1.3. Przepustowość magistrali 266
6.1.4. Przerwania: INTx#, PME#, MSI i SERR# 269

6.1.5. Bezpośredni dostęp do pamięci, emulacja DMA trybu ISA (PC/PCI, DDMA) ... 279
6.1.6. Mosty PCI i PCI-X ... 280
6.1.7. Konfiguracja urządzeń ... 292
6.1.8. Interfejs elektryczny i budowa magistrali ... 315
6.2. Interfejs AGP ... 325
6.2.1. Protokoły transakcji ... 327
6.2.2. Translacja adresów: szczelina AGP i GART ... 333
6.2.3. Transakcje izochroniczne AGP 3.0 ... 334
6.2.4. Rejestry konfiguracyjne AGP ... 335
6.2.5. Karty i gniazda AGP ... 339
6.3. Architektura PCI Express ... 343
6.3.1. Formaty transakcji i pakietów w architekturze PCI Express ... 349
6.3.2. Przesyłanie pakietów i przepustowość łączy ... 352
6.4. Interfejs LPC ... 353
6.5. Karty i magistrale rozszerzeń w notebookach ... 358
6.5.1. Interfejsy PCMCIA, karty PC i CardBus ... 359

Rozdział 7. Specjalizowane interfejsy urządzeń peryferyjnych ... 365
7.1. Interfejs klawiatury ... 365
7.1.1. Interfejsy klawiaturowe: AT i PS/2 ... 366
7.1.2. Kontroler klawiatury i myszy 8042 (8242) ... 368
7.1.3. Obsługa systemowa i interfejs programowy ... 371
7.2. Interfejs myszy ... 372
7.2.1. Myszy szeregowe: mysz Microsoft i mysz PC ... 372
7.2.2. Myszy PS/2 ... 373
7.2.3. Mysz szynowa ... 374
7.3. Interfejsy drukarki i plotera ... 375
7.3.1. Interfejsy równoległe: Centronics i IEEE 1284 ... 376
7.3.2. Szeregowe interfejsy drukarki ... 378
7.3.3. Obsługa drukarki w systemie operacyjnym ... 380
7.4. Interfejsy wyjściowe kart graficznych ... 381
7.4.1. Interfejs cyfrowy RGB TTL ... 381
7.4.2. Analogowe interfejsy RGB ... 382
7.4.3. Interfejsy cyfrowe P&D, DVI i DFP ... 389
7.4.4. Wewnętrzne interfejsy cyfrowe ... 393
7.4.5. Interfejsy sygnału wideo ... 395
7.5. Interfejsy audio ... 397
7.5.1. Interfejsy analogowe ... 397
7.5.2. Interfejsy cyfrowe ... 399
7.5.3. Interfejs MIDI ... 400
7.5.4. Interfejs karty syntezatora ... 403
7.6. Interfejs dżojstika ... 403

Rozdział 8. Interfejsy urządzeń pamięci masowych ... 407
8.1. Interfejs stacji dysków elastycznych ... 408
8.1.1. Interfejs sprzętowy ... 408
8.1.2. Kontroler stacji dysków elastycznych ... 411
8.2. Urządzenia IDE i interfejs ATA (ATAPI) ... 415
8.2.1. Interfejs równoległy ATA ... 418
8.2.2. Szeregowy interfejs ATA ... 446
8.3. Interfejsy i konstrukcja półprzewodnikowych pamięci masowych ... 452
8.3.1. CompactFlash ... 453
8.3.2. SmartMedia Card ... 456

8.3.3. Karty Multimedia Card i Secure Digital ... 457
8.3.4. Karty Miniature Card ... 459
8.4. Systemowa obsługa urządzeń pamięci masowych ... 460

Rozdział 9. Interfejsy sieciowe ... 463
9.1. Standardy i interfejsy sieci Ethernet ... 463
9.2. Adaptery sieciowe ... 468

Rozdział 10. Pomocnicze interfejsy i magistrale szeregowe ... 475
10.1. Kanały komunikacji szeregowej bazujące na I2C ... 476
10.1.1. Magistrala I2C ... 476
10.1.2. ACCESS.Bus ... 482
10.1.3. SMBus ... 488
10.1.4. Porównanie magistral I2C, ACCESS.Bus i SMBus ... 498
10.2. Media Independent Interface ... 499
10.3. Interfejs SPI ... 501
10.4. Interfejs JTAG ... 502
10.5. Programowe implementacje interfejsów szeregowych ... 507

Skorowidz ... 511

Rozdział 2.

Interfejs szeregowy: port COM

Uniwersalny interfejs szeregowy — port COM — występuje w komputerach PC począwszy od pierwszych modeli. Port zapewnia asynchroniczną[1] transmisję danych z wykorzystaniem standardu RS-232C. Porty COM są implementowane za pomocą układów scalonych określanych jako Uniwersalny Asynchroniczny Odbiorniko-nadajnik (ang. *Universal Asynchronous Receiver-Transmitter* — UART) zgodnych z modelem programowym *i8250/16450/16550*. Każdy z nich zajmuje osiem sąsiednich ośmiobitowych rejestrów w przestrzeni adresowej wejścia-wyjścia i może wykorzystywać jeden ze standardowych adresów: 3F8h (COM1), 2F8h (COM2), 3E8h (COM3) oraz 2E8h (COM4). Porty mogą generować przerwania sprzętowe IRQ4 (zwykle używane dla portów COM1 i COM3) oraz IRQ3 (używany dla COM2 i COM4). Ze strony sprzętowej porty zawierają linie do szeregowego nadawania danych i ich odbioru. Jest także zbiór linii sterujących i monitorowania stanu zgodnych ze standardem RS-232C. Porty COM są wyposażone w zewnętrzne męskie złącza *DB25P* lub *DB9P* na panelu tylnym komputera (patrz podrozdział 2.1). Cechą wyróżniającą interfejs jest zastosowanie sygnałów, które nie są sygnałami TTL: wszystkie zewnętrzne sygnały portu są bipolarne. Interfejs nie gwarantuje galwanicznej separacji: masa obwodu podłączonego urządzenia jest połączona z masą komputera. Szybkość transmisji sięga 115 200 b/s.

BIOS komputera obsługuje do czterech portów szeregowych: od COM1 do COM4 (komputery klasy AT zazwyczaj są wyposażone w dwa porty). Przerwanie BIOS `int14h` zapewnia inicjalizację portu, wejście i wyjście znaków (bez wykorzystania przerwań) i odpytywanie stanu. Za pomocą standardowych wywołań `int14h` można zaprogramować szybkości transmisji w zakresie od 110 do 9 600 b/s (mniej niż pozwalają rzeczywiste możliwości portu). W celu zwiększenia przepustowości powszechnie wykorzystuje się interakcję oprogramowania aplikacyjnego z portem na poziomie rejestrów. Z tego powodu sprzęt portów szeregowych musi być zgodny z modelem oprogramowania i8250/1645/16550.

[1] W komputerach PC transmisję synchroniczną obsługują tylko specjalne adaptery, takie jak SDLC lub V.35.

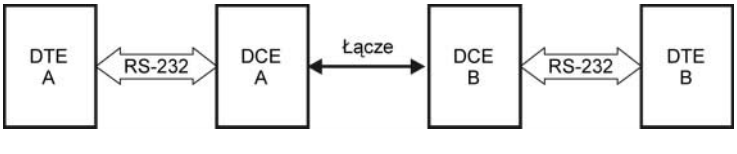
Nazwa portu wskazuje na jego główne przeznaczenie: podłączanie sprzętu (np. modemów) umożliwiającego komunikację z innymi komputerami, sieciami i urządzeniami peryferyjnymi. Urządzenia peryferyjne wyposażone w interfejsy szeregowe — drukarki, plotery, itp. można podłączać bezpośrednio do portu. Porty COM są powszechnie wykorzystywane do podłączania myszy oraz do bezpośredniego podłączania dwóch komputerów. Porty COM wykorzystuje się również w celu podłączania do komputera kluczy elektronicznych.

Praktycznie wszystkie nowoczesne płyty główne (począwszy od pierwszych płyt wyposażonych w złącze PCI dla procesorów x486) są wyposażone w dwa wbudowane adaptery portu COM. Jeden z portów można także wykorzystać do bezprzewodowej komunikacji z urządzeniami peryferyjnymi za pomocą podczerwieni (IrDA). Istnieją karty ISA wyposażone w kilka portów COM. Najczęściej obok portów COM na kartach tego typu jest port LPT oraz kontrolery interfejsów dysków (FDC+IDE). Jeśli jest potrzebna większa liczba interfejsów, można zainstalować dodatkowe specjalizowane adaptery nazywane kartami wieloportowymi. Są to stosunkowo drogie karty wyposażone w 4, 8, 12, a nawet 16 portów. Wyprowadzenie tak dużej liczby złączy na tylnym panelu komputera PC jest kłopotliwe. Z tego powodu karty wieloportowe są zazwyczaj wyposażone w blok zewnętrznych złączy zawierających elektronikę i podłączonych do adaptera za pomocą kabla z wielożyłowymi złączami. Karty wieloportowe nie są obsługiwane przez BIOS.

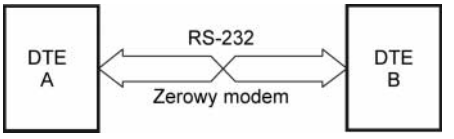
„Klasyczny" port COM umożliwiał jedynie transmisję danych sterowaną programowo. W celu przesłania każdego bajtu procesor musiał wykonać kilka instrukcji. Nowoczesne porty są wyposażone w kilka buforów danych FIFO i umożliwiają transfer danych poprzez kanał DMA. To znacznie zmniejsza obciążenie procesora — szczególnie istotny czynnik przy dużych szybkościach transmisji.

2.1. Interfejs RS-232C

Interfejs RS-232C jest przeznaczony do podłączania sprzętu do transmisji lub odbierania danych (ang. *Data Terminal Equipment* — DTE) ze sprzętem zapewniającym kanały danych (ang. *Data Communication Equipment* — DCE). Rolę urządzenia DTE może spełniać komputer, drukarka, ploter lub inne urządzenie peryferyjne. Rolę urządzenia DCE zwykle spełnia modem. Celem połączenia jest zapewnienie łącza pomiędzy dwoma urządzeniami transmisji danych. Schemat połączeń pokazano na rysunku 2.1. Interfejs umożliwia wyłączenie kanału zdalnej komunikacji wraz z parą urządzeń DCE poprzez bezpośrednie połączenie dwóch urządzeń za pomocą kabla modemu zerowego (ang. *null-modem*) (rysunek 2.2).



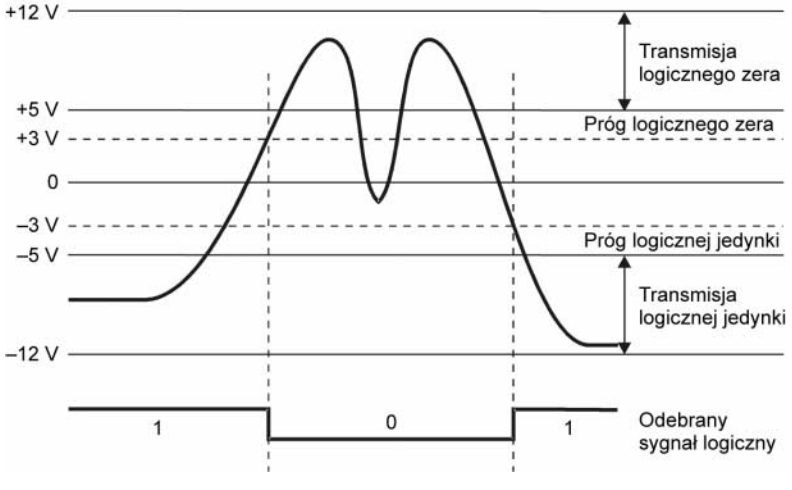
Rysunek 2.1. *Schemat połączeń interfejsu RS-232C*



Rysunek 2.2. *Połączenie RS-232C za pomocą kabla zerowego modemu*

W standardzie określono sygnały sterujące interfejsu, transmisji danych, interfejsu elektrycznego i typy złączy. Standard zapewnia asynchroniczne i synchroniczne tryby transmisji, ale porty COM obsługują jedynie tryb asynchroniczny. Funkcjonalnie standard RS-232C odpowiada standardom CCITT/ITU V.24/V.28, ale występują w nim inne nazwy sygnałów.

W standardzie RS-232C (patrz rysunek 2.3) zdefiniowano asymetryczne nadajniki i odbiorniki, w których sygnały są przesyłane względem wspólnego przewodu: obwodu masy (w innych interfejsach, takich jak RS-422, są wykorzystywane symetryczne sygnały różnicowe). Interfejs nie gwarantuje galwanicznej separacji urządzeń. Napięcia w zakresie –12 V do –3 V odpowiadają logicznej jedynce (stan *MARK*) na linii danych odbiornika (sygnał `RXD`). Napięcia w zakresie +3 V do +12 V odpowiadają logicznemu zeru (stan *SPACE*). Dla sygnałów sterujących napięcia w zakresie od +3 V do +12 V odpowiadają stanowi *ON*, natomiast w zakresie –12 V do –3 V stanowi *OFF*. Zakres od –3 V do +3 V to martwa strefa zapewniająca histerezę odbiornika: stan linii uznaje się za zmieniony wyłącznie po przekroczeniu progu (rysunek 2.3). Poziomy sygnałów na wyjściach nadajnika powinny mieścić się w zakresie od –12 V do –5 V oraz od +5 V do +12 V. Różnica potencjałów pomiędzy masami sygnałów (SG) połączonych urządzeń powinna być mniejsza niż 2 V. Większa różnica potencjałów może spowodować błędny odbiór sygnałów. Warto zwrócić uwagę, że poziomy sygnałów TTL (na wejściach i wyjściach układów scalonych UART) są przesyłane z wykorzystaniem prostego kodowania na liniach `TXD` i `RXD` oraz odwróconego dla pozostałych.



Rysunek 2.3. *Odbiór sygnałów RS-232C*

Interfejs zakłada istnienie uziemienia zabezpieczającego dla podłączonych urządzeń, jeśli oba są zasilane prądem zmiennym z sieci i są wyposażone w filtry.



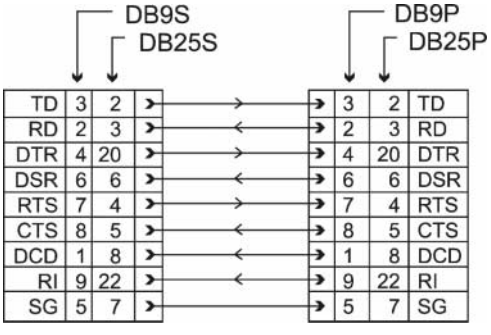
Podłączanie i odłączanie kabli interfejsów urządzeń wyposażonych w indywidualne zasilacze powinno odbywać się przy wyłączonym zasilaniu. W innym przypadku różnica napięć w nieustabilizowanych urządzeniach w momencie nawiązywania połączenia może spowodować przepływ prądu w obwodach wyjściowych lub, co jest bardziej niebezpieczne, w obwodach wejściowych interfejsu i uszkodzić układy scalone.

Standard RS-232C reguluje *typy wykorzystywanych złączy*.

Zazwyczaj urządzenia DTE (włącznie z portami COM) są wyposażone w męskie złącze DB-25P lub mniejsze męskie złącze DB-9P. Złącza dziewięciostykowe nie są wyposażone w dodatkowe sygnały potrzebne do obsługi trybu synchronicznego (w większości złączy dwudziestopięciostykowych styki te są nieużywane).

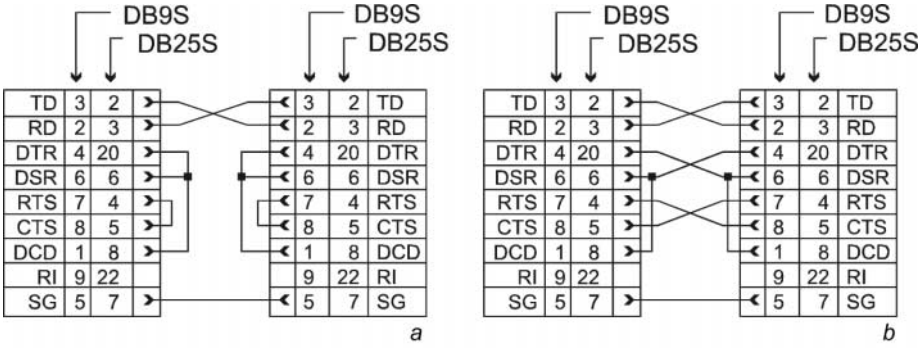
Urządzenia DCE (modemy) są wyposażone w żeńskie złącza *DB-25S* lub *DB-9S*.

Z reguły tej wynika, że złącza urządzeń DCE można połączyć ze złączami urządzeń DTE bezpośrednio lub za pomocą kabli adapterów „na wprost" wyposażonych w żeńskie złącze z jednej strony i męskie z drugiej, których przewody są połączone „jeden do jednego". Istnieją też kable adapterów wyposażone w złącze 9-stykowe z jednej strony i 25-stykowe z drugiej (rysunek 2.4).



Rysunek 2.4. *Kable do podłączania modemów*

Jeśli urządzenia DTE są podłączone bez wykorzystania modemów, ich złącza są połączone za pomocą *kabla zerowego modemu* (Z-modem). Takie kable są wyposażone w żeńskie złącza po obu stronach, a żyły skrzyżowane są zgodnie ze schematem pokazanym na rysunku 2.5.



Rysunek 2.5. *Kable zerowego modemu: a — minimalny, b — kompletny*

Jeśli w urządzeniu DTE zainstalowano złącze żeńskie, istnieje prawie stuprocentowa pewność, że należy je podłączyć do innego urządzenia za pomocą kabla na wprost podobnego do kabla wykorzystywanego do połączenia modemu. Żeńskie złącza zazwyczaj montuje się w urządzeniach, których nie łączy się za pośrednictwem modemów.

W tabeli 2.1 zaprezentowano funkcje styków złącza portu COM (oraz innych urządzeń typu DTE). Styki złącza DB-25S są zdefiniowane przez standard EIA/TIA-232-E, natomiast złącza DB-9S przez standard EIA/TIA-574. W modemach (DCE) stosuje się te same nazwy obwodów i styków, ale funkcje sygnałów (wejście-wyjście) są odwrócone.

Tabela 2.1. *Złącza interfejsu RS-232C i wykorzystywane w nich sygnały*

Sygnał			Styk złącza		Numer żyły zewnętrznego kabla łączącego z komputerem PC				Kierunek Wejście/Wyjście
Port COM	RS-232 V.24		DB-25P	DB-9P	1[1]	2[2]	3[3]	4[4]	
PG	AA	101	1	5	(10)	(10)	(10)	1	—
SG	AB	102	7	5	5	9	1	13	—
TD	BA	103	2	3	3	5	3	3	Wyjście
RD	BB	104	3	2	2	3	4	5	Wejście
RTS	CA	105	4	7	7	4	8	7	Wyjście
CTS	CB	106	5	8	8	6	7	9	Wejście
DSR	CC	107	6	6	6	2	9	11	Wejście
DTP	CD	108/2	20	4	4	7	2	14	Wyjście
DCD	CF	109	8	1	1	1	5	15	Wejście
RI	CE	125	22	9	9	8	6	18	Wejście

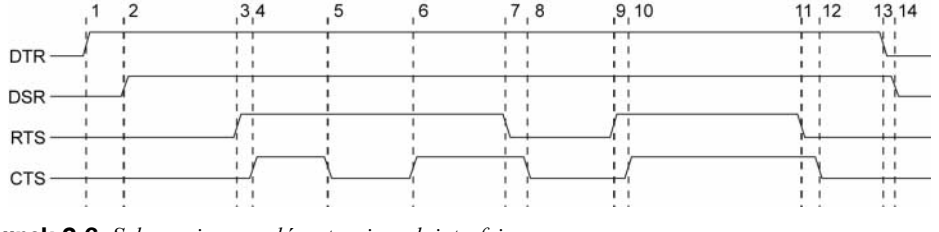
[1] *Taśma dla ośmiobitowych kart wielofunkcyjnych.*

[2] *Taśma dla szesnastobitowych kart wielofunkcyjnych i portów wbudowanych.*

[3] *Wersja taśmy dla portów wbudowanych.*

[4] *Szeroka taśma dla złącza 25-stykowego.*

Poniżej przeanalizujemy podzbiór sygnałów interfejsu *RS-232C* dla trybu asynchronicznego z punktu widzenia portu COM komputera PC. Dla wygody wykorzystamy mnemoniki przyjęte w opisach portów COM i większości urządzeń (które różnią się od anonimowych notacji opisu kabli RS-232 i V.24). Należy pamiętać, że stan aktywny (ON) sygnałów sterujących i logiczne zero przesyłanych danych są reprezentowane przez potencjał dodatni (powyżej +3 V) interfejsu, natomiast stan OFF i logiczna jedynka są reprezentowane przez potencjał ujemny (poniżej –3 V). W tabeli 2.2 zaprezentowano funkcje sygnałów interfejsu. Na rysunku 2.6 pokazano normalną sekwencję sygnałów sterujących dla przypadku modemu podłączonego do portu COM:



Rysunek 2.6. *Sekwencja sygnałów sterujących interfejsu*

Tabela 2.2. *Funkcje sygnałów interfejsu RS-232C*

Sygnał	Funkcja
PG	*Uziemienie zabezpieczające*: podłączone do obudowy urządzenia i ekranu kabla.
SG	*Masa sygnału (obwodu)*: w zależności od poziomu sygnału.
TD	*Transmisja danych*: dane przesyłane szeregowo, wynik generowany przez nadajnik.
RD	*Odbiór danych*: dane szeregowe, dane wejściowe odbiornika.
RTS	*Request To Send* (żądanie wysłania): stan ON informuje modem, że terminal posiada dane do transmisji. W trybie półdupleksowym wykorzystywany do sterowania kierunkiem transmisji: stan ON to sygnał dla modemu do przełączenia do trybu transmisji.
CTS	*Clear To Send* (zezwolenie na wysłanie): zezwolenie wysyłania danych dla terminala. Stan OFF uniemożliwia transmisję danych. Sygnał jest wykorzystywany do sprzętowej kontroli strumieni danych.
DSR	*Data Set Ready* (zbiór danych gotowy): odebranie sygnału oznacza, że urządzenie jest gotowe do transmisji danych (modem w trybie działania jest podłączony do kanału i zakończył negocjacje z urządzeniem po drugiej stronie kanału).
DTR	*Data Terminal Ready* (terminal danych gotowy): wysłanie sygnału oznacza, że terminal jest gotowy do wymiany danych. Stan ON utrzymuje połączenie komutowanego kanału.
DCD	*Data Carrier Detected* (wykryto nośną danych): odebranie sygnału oznacza, że wykryto sygnał nośnej zdalnego modemu.
RI	*Ring Indicator* (wskaźnik dzwonka): odebranie sygnału wywołania (dzwonka). W komutowanym kanale sygnał ten oznacza, że modem otrzymuje sygnał dzwonienia.

1. Komputer PC sygnalizuje, że chce skorzystać z modemu, poprzez ustawienie linii DTR.
2. Modem sygnalizuje, że jest gotowy i ustanawia połączenie, poprzez ustawienie sygnału DSR.
3. Poprzez ustawienie sygnału RTS komputer prosi o zezwolenie na transmisję i informuje modem, że jest gotowy do odbioru danych.
4. Poprzez ustawienie sygnału CTS modem informuje, że jest gotowy do odbioru danych z komputera i do wysyłania ich w łączu.
5. Poprzez wyzerowanie sygnału CTS modem sygnalizuje, że nie może kontynuować odbioru danych (na przykład bufor jest pełny) oraz że komputer powinien wstrzymać wysyłanie danych.
6. Poprzez ustawienie sygnału CTS modem zezwala na wznowienie transmisji danych (zwolniło się miejsce w buforze).
7. Wyzerowanie sygnału RTS może oznaczać, że bufor komputera jest pełny (modem musi wstrzymać wysyłanie danych do komputera) lub że komputer nie ma danych do przesłania do modemu. W takim przypadku modem zwykle wstrzymuje transmisję danych do komputera.
8. Modem potwierdza wyzerowanie sygnału RTS poprzez wyzerowanie sygnału CTS.
9. W celu wznowienia wysyłania danych komputer ponownie ustawia sygnał RTS.

10. Modem potwierdza gotowość.
11. Komputer sygnalizuje zakończenie wymiany.
12. Modem odpowiada, wysyłając potwierdzenie.
13. Komputer zeruje sygnał DTR, co zazwyczaj oznacza sygnał do przerwania połączenia.
14. Poprzez wyzerowanie sygnału DSR modem sygnalizuje przerwanie połączenia.

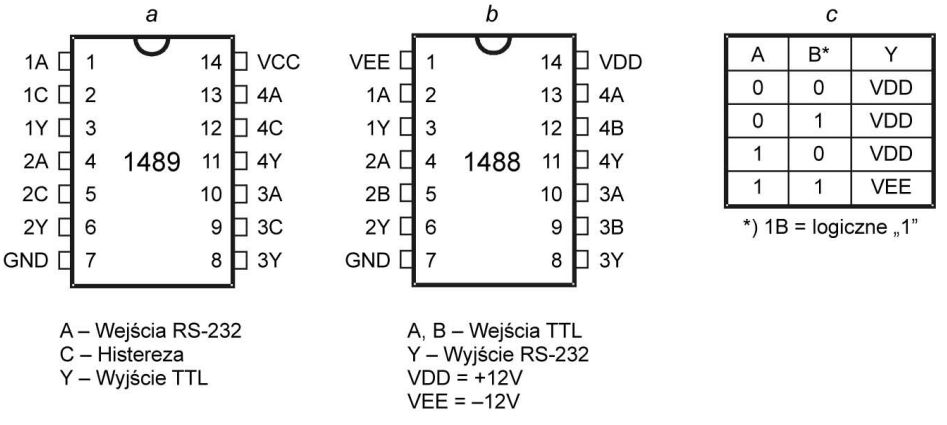
Z analizy powyższej sekwencji jasno wynika cel połączenia sygnałów DTR/SR oraz RTS/CTS w kablu zerowego modemu.

2.2. Interfejsy pomocnicze i konwertery poziomów

W interfejsach szeregowych nie zawsze wykorzystuje się bipolarne sygnały RS-232C. Korzystanie z nich jest niewygodne np. z powodu konieczności wykorzystania bipolarnych zasilaczy dla nadajniko-odbiorników. Układy scalone opisanych powyżej nadajniko-odbiorników UART pracują z sygnałami TTL(CMOS). Te same rodzaje sygnałów są stosowane w interfejsach różnorodnych urządzeń. Na przykład sygnały TTL(CMOS) są wykorzystywane w portach twardych dysków, telefonów komórkowych (choć o bardzo niskich poziomach) oraz różnych innych urządzeniach. Wiele urządzeń (włącznie z kieszonkowymi komputerami PC i telefonami komórkowymi) jest wyposażonych w interfejsy szeregowe wykorzystujące logikę niskonapięciową. Oczywiście sygnały standardowej logiki nie charakteryzują się tak wysoką rezystancją szumów jak RS-232C, ale nie zawsze jest to potrzebne.

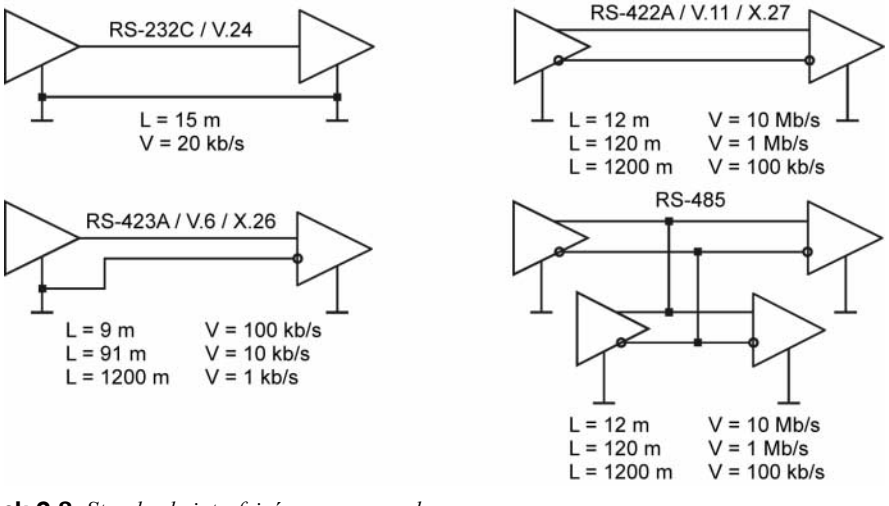
W celu umożliwienia wzajemnej konwersji pomiędzy poziomami napięć interfejsu RS-232C i logiki CMOS istnieją specjalne układy scalone buforów dla odbiorników (z histerezą) oraz nadajników sygnałów bipolarnych. Układy te jako pierwsze stają się ofiarami efektów „pirotechnicznych" w przypadku naruszenia reguł uziemiania i łączenia urządzeń. Dawniej często instalowano je w podstawkach w celu ułatwienia wymiany w przypadku awarii. Na rysunku 2.7 zaprezentowano rozmieszczenie styków jednego z popularnych układów scalonych generujących sygnały na poziomie interfejsu RS-232C. Często obwody buforów są włączone w układy scalone LSI interfejsów. W ten sposób cena urządzenia staje się niższa, ale w przypadku awarii powstaje ryzyko większych strat. Uszkodzenie układów scalonych interfejsu z powodu zwarcia jest mało prawdopodobne: prąd zwarcia nadajników zazwyczaj nie przekracza 20 mA.

W specjalnych kablach adapterów często stosowane są konwertery produkowane przez firmy Maxim i Sypex. Kable te sprawdzają się pod tym względem, że zawierają zarówno odbiorniki, jak nadajniki. Spośród szerokiej gamy dostępnych konwerterów z łatwością można wybrać te, które zawierają potrzebną liczbę odbiorników i nadajników, a także wymaganą charakterystykę napięciową (unipolarne, bipolarne, niskonapięciowe).



Rysunek 2.7. *Generatory sygnałów RS-232C: a — odbiornik 1489; b — odbiornik 1488; c — tabela stanów wyjściowych nadajnika (*na brakującym wyjściu 1B zakłada się logiczną jedynkę)*

Kiedy jest wymagana wysoka rezystancja szumów (dla dużych szybkości lub transmisji na dalekie odległości), wykorzystywane są różne wersje elektryczne interfejsu szeregowego: *RS-422A (V.11, X.27)*, *RS-423A (V.10, X.26)* oraz *RS-485*. Na rysunku 2.8 pokazano schematy podłączania odbiorników i nadajników wraz z limitami długości łączy (L) oraz maksymalnymi szybkościami transmisji (V). Asymetryczne linie interfejsów *RS-232C* i *RS-423A* charakteryzują się najniższą odpornością na znane zakłócenia, chociaż różnicowe wejście odbiornika *RS-423A* w pewnym stopniu neutralizuje tę sytuację. Najlepsze parametry mają interfejsy *RS-422A* i *RS-485* pracujące w symetrycznych liniach komunikacyjnych. Wykorzystuje się w nich nadajniki różnicowe z osobną parą przewodów (z przeplotem) dla każdej pętli sygnału.



Rysunek 2.8. *Standardy interfejsów szeregowych*

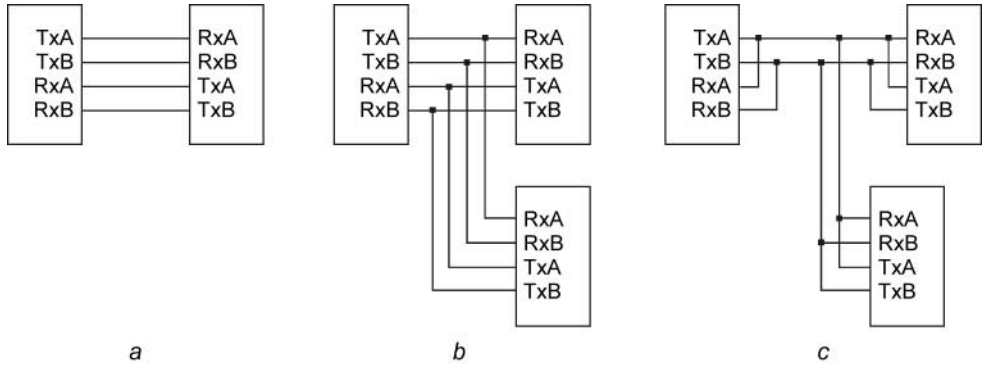
W interfejsach *EIA-RS-422* (ITU-T V.11, X.27) i *EIA-RS-485* (ISO 8482) wykorzystuje się symetryczną transmisję sygnału. Interfejsy te umożliwiają realizację połączeń zarówno w topologii punkt-punkt, jak i w magistrali. Nośnikiem informacji jest

w nich różnica potencjałów pomiędzy przewodnikami A i B. Różnica potencjałów na wejściu odbiornika $U_A–U_B > 0{,}2$ V (A ma większy potencjał dodatni niż B) odpowiada stanowi OFF (przerwa), natomiast $U_A–U_g < -0{,}2$ V (A ma większy potencjał ujemny niż B) stanowi ON (mark). Zakres $|U_A–U_B|<0{,}2$ V to martwa strefa (histereza) zapewniająca ochronę przed zakłóceniami. Na wyjściach nadajnika sygnały U_A i U_g zazwyczaj mają poziomy 0 V i +5 V (CMOS) lub +1 V i +4 V (TTL). Różnicowe napięcie wyjściowe musi mieścić się w zakresie od 1,5 V do 5 V. Rezystancja wyjściowa nadajnika wynosi 100 Ω. Interfejsy są zgodne pod względem elektrycznym, chociaż występują pewne różnice w ograniczeniach. Zasadniczą cechą wyróżniającą nadajniki RS-485 jest ich zdolność do przełączania się do stanów wysokiej impedancji. Nadajniki RS-422/ 485 są zgodne z odbiornikami RS-423. W tabeli 2.3 wyszczególniono główne parametry interfejsów. Topologie połączeń pokazano na rysunku 2.9.

Tabela 2.3. *Parametry interfejsów RS-422 i RS-485*

Parametr	RS-422	RS-485
Próg działania |UA–UB| [V]	0,2	0,2
Dopuszczalny poziom zakłóceń trybu wspólnego, [V][1].	–6.8 – +6.8	–6.8 – +11.8
Dopuszczalne napięcie wejściowe, [V][1].	–7 – +7	–7 – +12
Rezystancja wejściowa odbiornika, [kΩ].	4	12
Minimalna rezystancja obciążenia nadajnika, [Ω].	100	60
Maksymalna liczba węzłów.	Jeden nadajnik plus 10 odbiorników	32 (nadajniki, odbiorniki lub kombinacja nadajników i odbiorników)
Maksymalna długość [m].	1200 (100 kb/s) 12 (10 Mb/s)	1200 (100 kb/s) 12 (10 Mb/s)
Terminator, R=100 Ω.	Po stronie nadajnika	Po obu stronach
Prąd zwarcia [mA].	<150 do szyny GND	<250 do szyny o potencjale –7 V – +12 V lub pomiędzy żyłami A i B

[1] *Napięcie jest mierzone względem masy obwodu węzła.*



Rysunek 2.9. *Topologia interfejsów: a — RS-422; b — RS-485 4-żyłowy; c — RS-485 2-żyłowy*

W celu zwiększenia liczby węzłów można podnieść rezystancję wejściową odbiorników, ale to prowadzi do obniżenia dopuszczalnej szybkości transmisji lub zmniejszenia potencjalnej odległości, na którą jest realizowana transmisja. Maksymalna szybkość transmisji na krótkich odległościach (do 10 metrów) jest ograniczona szybkością działania nadajnika (można osiągnąć częstotliwości do 25 MHz). Na średnich odległościach ograniczenia zależą od pojemności kabla (przy szybkości 1 200 b/s — 25 nF, przy szybkości 9 600 b/s — 30 nF, przy szybkości 115 kb/s — 250 pF). Maksymalna odległość jest ograniczona rezystancją DC pętli.

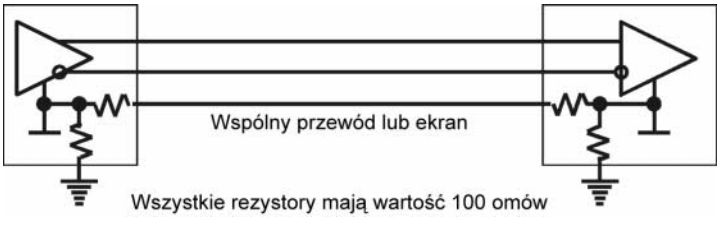
Istnieją dwa typy interfejsu RS-485: dwużyłowy i czterożyłowy. Interfejs czterożyłowy (rysunek 2.9b) ustawia węzeł nadrzędny, którego nadajniki obsługują odbiorniki pozostałych węzłów. Nadajnik węzła nadrzędnego jest zawsze aktywny: nie musi się przełączać do stanu wysokiej impedancji. Nadajniki pozostałych węzłów (podrzędnych) muszą posiadać trójstanowe wyjścia — są podłączone do wspólnej szyny nadajnika nadrzędnego. W interfejsie dwużyłowym (rysunek 2.9c) wszystkie węzły są sobie równe.

W najprostszym przypadku połączenia punkt-punkt interfejsy RS-485 i RS-422 są równoważne i stan wysokiej impedancji nie jest używany.

W celu utrzymania najmniej aktywnego stanu szyny RS-485, kiedy nie ma aktywnych nadajników, na linii instaluje się aktywny terminator, który utrzymuje potencjał przewodników. W tym stanie żyła B musi charakteryzować się wyższym potencjałem dodatnim niż żyła A.

W przypadku połączenia wielopunktowego należy zapewnić metodę dostępu do środowiska transmisji. Najczęściej wykorzystuje się metodę odpytywania: wyznaczone urządzenie nadrzędne odpytuje urządzenia podrzędne, sprawdzając ich gotowość do transmisji. Przekazanie praw dostępu z jednego urządzenia do innego jest możliwe zgodnie z ustanowionym protokołem. Czasami stosuje się także metody dostępu swobodnego (analogicznie do sieci Ethernet).

Wejście różnicowe interfejsów zapewnia zabezpieczenie przed zakłóceniami, ale w tym przypadku masy obwodów urządzeń muszą być połączone między sobą oraz do szyny uziemienia. W celu połączenia urządzeń wykorzystywana jest trzecia żyła interfejsu (do tego celu można także wykorzystać ekran kabla). W celu zabezpieczenia przed przepływem przez trzecią żyłę zbyt dużego prądu równoważącego potencjały masy, w obwodzie umieszcza się rezystory (rysunek 2.10).



Rysunek 2.10. *Połączenia masy obwodów dla interfejsów RS-422 i RS-485*

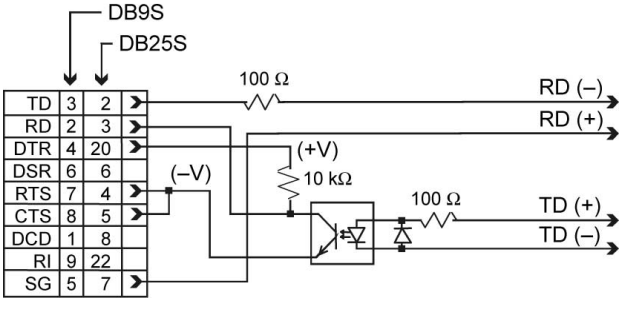
Interfejs RS-422 jest często wykorzystywany do podłączania urządzeń peryferyjnych (na przykład drukarek). Interfejs RS-485 to popularny sposób podłączenia sprzętu automatyki przemysłowej.

W celu transmisji sygnału w dwużyłowej linii łączącej odbiornik z nadajnikiem stosowany jest *interfejs pętli prądowej*, w którym zamiast napięciowych stosowane są sygnały prądowe. Prąd o natężeniu 20 mA odpowiada logicznej jedynce, natomiast brak przepływu prądu to logiczne zero. Taki sposób reprezentacji sygnałów w opisanym powyżej formacie transmisji asynchronicznej umożliwia wykrywanie przerw w łączu: odbiornik wykryje brakujący bit stopu (przerwa w łączu spełnia rolę stałego logicznego zera).

Pętla prądowa zazwyczaj oznacza *galwaniczną separację* obwodów wejściowych odbiornika od pozostałej części obwodu. W tym przypadku źródłem prądu w pętli jest nadajnik (w tej wersji nazywa się aktywnym nadajnikiem). Możliwe jest również zasilanie z odbiornika (aktywny odbiornik). W takim przypadku wyjście nadajnika może być galwanicznie odseparowane od pozostałej części obwodu nadajnika. Istnieją również wersje uproszczone, które nie zapewniają galwanicznej separacji, ale są to zubożone wersje interfejsu. Warto zwrócić uwagę, że interfejs MIDI (patrz punkt 7.5.3) nie jest zgodny z „klasyczną" pętlą prądową.

Pętla prądowa z galwaniczną separacją (rysunek 2.11) umożliwia transmisję sygnałów do kilku kilometrów, ale z niskimi szybkościami (szybkości powyżej 19 200 b/s nie są wykorzystywane, a kiedy odległość mierzy się w kilometrach, maksymalna szybkość wynosi 9 600 b/s, a nawet mniej). Maksymalna odległość jest określona przez rezystancję pary przewodów oraz poziom szumów. Ponieważ interfejs wymaga jednej pary przewodów dla każdego sygnału, zazwyczaj używane są tylko dwa sygnały interfejsu szeregowego (linia czterożyłowa). W przypadku wymiany dwukierunkowej wykorzystywane są wyłącznie sygnały do wysyłania i odbioru danych. W takiej sytuacji stosuje się programowe sterowanie przepływem XON/XOFF. Jeśli nie ma potrzeby dwukierunkowej wymiany, jedna para przewodów jest wykorzystywana do transmisji danych, a druga do sterowania przepływem za pomocą sygnału CTS (protokół sprzętowy) lub odwróconej linii danych (protokół programowy). Przy pomocy odpowiedniego oprogramowania można wykorzystać jedną pętlę prądową do zorganizowania półdupleksowej dwukierunkowej komunikacji pomiędzy dwoma urządzeniami. W takim przypadku każdy odbiornik „słyszy" zarówno sygnały nadajników z drugiej strony łącza, jak sygnały nadajników własnych. Protokoły komunikacyjne uznają je jako sygnały echa. Aby był możliwy odbiór bez błędów, nadajniki muszą pracować na zmianę.

Rysunek 2.11. *Konwersja interfejsu RS-232C na pętlę prądową*

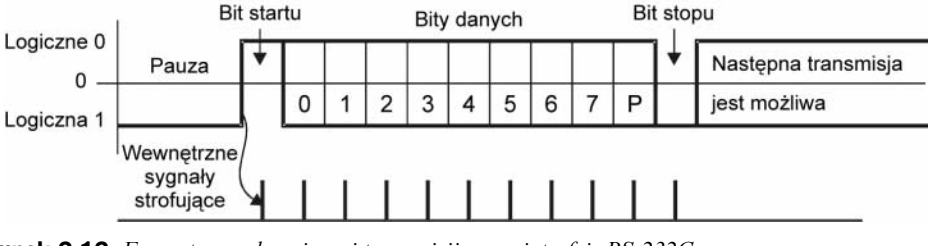


Pętla prądowa umożliwia wykorzystanie dedykowanych fizycznych łączy (dzierżawionych) bez modemów, ale z niskimi szybkościami transmisji. Czasami pętle prądowe stosuje się do podłączenia terminali interfejsu RS-232C w przypadku, kiedy odległość połączenia gwarantowana przez interfejs jest zbyt mała lub jeśli jest potrzebna separacja galwaniczna. Przekształcenie sygnałów RS-232C na sygnały pętli prądowej nie jest

trudne. Na rysunku 2.11 pokazano bardzo prosty schemat konwertera umożliwiającego podłączenie terminala. Do uzyskania sygnałów bipolarnych wymaganych przez obwody wejściowe portu COM wykorzystywane jest zasilanie interfejsu. Konwerter można usprawnić, zapewniając ochronę przed przeciążeniem optoizolatorów oraz w celu stabilizacji sygnałów napięciowych. Maksymalna szybkość transmisji zależy od szybkości wykorzystanych optoizolatorów (szybkość 9 600 b/s można osiągnąć stosując optoizolatory dowolnego typu).

2.3. Tryb asynchroniczny

Tryb asynchroniczny transmisji szeregowej jest bajtowy (znakowy): najmniejsza jednostka przesyłanych informacji to jeden bajt (jeden znak). Na rysunku 2.12 zilustrowano format transmisji bajtowej. Transmisja każdego bajtu rozpoczyna się od *bitu startu*, który sygnalizuje początek transmisji do odbiornika. Po nim są przesyłane *bity danych* i, czasami, *bit parzystości*. Transmisję kończy *bit stopu*. *Bit startu* kolejnego bajtu jest przesyłany w dowolnym czasie po bicie stopu poprzedniego bajtu, tzn. pomiędzy transmisjami poszczególnych bajtów mogą występować pauzy dowolnej długości. Zastosowanie bitu startu, który zawsze ma ściśle zdefiniowaną wartość (logiczne zero), to prosta metoda synchronizacji odbiornika za pomocą sygnału z nadajnika. Odbiornik i nadajnik pracują z tą samą szybkością transmisji. Wewnętrzny generator synchronizacji odbiornika wykorzystuje licznik dzielnika częstotliwości, który jest ustawiany na wartość zero w momencie odebrania bitu startu. Licznik ten generuje wewnętrzne sygnały strobujące, dzięki którym odbiornik odbiera następne bity. W idealnej sytuacji impulsy strobujące są generowane w środku interwałów bitowych, co umożliwia odbiór danych nawet wtedy, gdy szybkości transmisji nadajnika i odbiornika nieznacznie się różnią. Jest oczywiste, że w przypadku transmisji ośmiu bitów danych, jednego bitu parzystości i jednego bitu stopu, dopuszczalna tolerancja częstotliwości zapewniająca prawidłowy odbiór danych nie może przekroczyć pięciu procent. Biorąc pod uwagę zniekształcenia fazy oraz kwantową naturę licznika synchronizacji, dopuszczalna tolerancja częstotliwości jest jeszcze mniejsza. Im mniejszy współczynnik podziału wewnętrznego generatora częstotliwości (wyższa częstotliwość transmisji), tym mniej dokładne umieszczenie sygnału strobującego w stosunku do środka przedziału bitowego, a zatem większe wymagania dopasowania częstotliwości.



Rysunek 2.12. *Format asynchronicznej transmisji przez interfejs RS-232C*

Im wyższa częstotliwość transmisji, tym bardziej zniekształcone sygnały wpływają na fazę odbieranych sygnałów. Połączenie tych czynników skutkuje zwiększeniem wymagań dotyczących dopasowania częstotliwości odbiornika i nadajnika w miarę zwiększania się szybkości transmisji.

Format transmisji asynchronicznej umożliwia wykrywanie potencjalnych *błędów transmisji*.

- Jeśli otrzymano sygnał rozpoczęcia transmisji, ale w momencie strobowania bitu startu odebrano logiczną jedynkę, bit startu można uznać za błędny. W takiej sytuacji odbiornik powraca do stanu oczekiwania. Odbiornik nie musi zgłaszać tego błędu.
- Jeśli podczas przedziału czasu przydzielonego dla bitu stopu zostanie wykryte logiczne zero (poziom wysoki), następuje zarejestrowanie bitu stopu.
- W przypadku wykorzystania kontroli parzystości, po wysłaniu bitów danych wysyłany jest bit parzystości. Bit ten uzupełnia liczbę logicznych jedynek w bitach danych do liczby parzystej lub nieparzystej, w zależności od wcześniejszych ustawień. Błąd jest rejestrowany w przypadku odebrania bajtu zawierającego niepoprawną wartość bitu parzystości.
- Kontrola formatu umożliwia wykrycie przerw łącz: z reguły w przypadku przerwy w łączu odbiornik „widzi" logiczne zero. To logiczne zero początkowo jest interpretowane jako bit startu lub zerowe bity danych, ale później wykonywana jest kontrola bitu stopu.

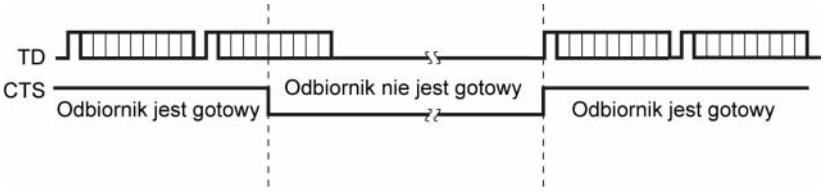
Dla trybu asynchronicznego przyjmuje się zbiór *standardowych szybkości transmisji*: 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 i 115200 bitów na sekundę (b/s). Czasami zamiast terminu b/s wykorzystuje się termin bod, ale dla transmisji danych binarnych określenie to jest niepoprawne. Body wykorzystuje się do pomiaru częstotliwości zmian stanu łącza, ale w przypadku zastosowania w kanale komunikacji niebinarnej metody kodowania (którą często wykorzystuje się w nowoczesnych modemach), szybkość transmisji bitów (b/s) oraz częstotliwość zmiany sygnału (liczba bodów) mogą się znacznie różnić.

Liczba *bitów danych* może wynosić 5, 6, 7 lub 8 (formaty pięcio- lub sześciobitowe wykorzystuje się bardzo rzadko). Liczba *bitów stopu* może wynosić 1, 1,5 lub 2 (półtora bitu oznacza jedynie czas trwania przerwy).

2.4. Sterowanie przepływem danych

Do sterowania przepływem danych można wykorzystać dwa rodzaje protokołów: sprzętowy lub programowy. Czasami sterowanie przepływem jest mylone z uzgadnianiem lub potwierdzaniem. *Uzgadnianie* (ang. *handshaking*) oznacza negocjowanie z partnerem wszystkich elementarnych kroków protokołu (dobrym przykładem jest sygnalizacja protokołów transmisji portu równoległego w trybach EPP i ECP). *Potwierdzanie* to przesyłanie powiadomienia o odebraniu porcji informacji (bajtu, ramki lub pakietu). Sterowanie przepływem zakłada wysyłanie powiadomień dotyczących tego, czy jest możliwy dalszy odbiór danych. Dość często sterowanie przepływem wykorzystuje mechanizmy uzgadniania lub potwierdzania.

W *sprzętowym protokole sterowania przepływem RTS/CTS* wykorzystuje się sygnał `CTS` umożliwiający wstrzymanie wysyłania danych, jeśli odbiornik nie jest gotowy do ich odbierania (rysunek 2.13).



Rysunek 2.13. *Sprzętowe sterowanie przepływem*

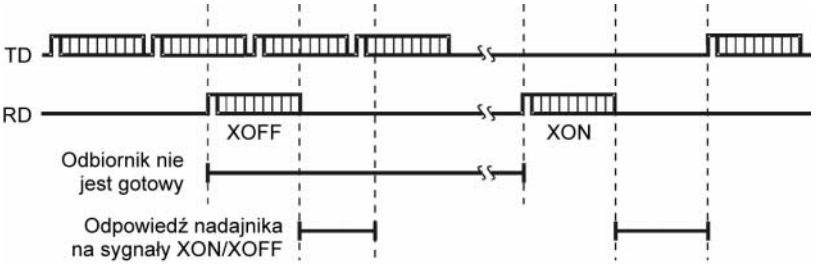
Nadajnik umożliwia wysłanie następnego bajtu tylko wtedy, kiedy jest ustawiona linia CTS. Zatrzymanie bajtu, którego transmisja rozpoczęła się za pomocą sygnału CTS, jest niemożliwe (co zapewnia integralność transmisji). Protokół sprzętowy zapewnia najszybszą reakcję nadajnika na stan odbiornika. Układy scalone asynchronicznych nadajniko-odbiorników zawierają co najmniej dwa rejestry odbiorcze: rejestr przesunięcia, umożliwiający odbiór następnego pakietu i rejestr pamięci, z którego jest odczytywany odebrany bajt. Dzięki temu można zaimplementować protokół sprzętowy bez utraty danych.

Protokół sprzętowy sprawdza się do podłączania drukarek i ploterów, jeśli urządzenia te go obsługują. Kiedy dwa komputery są połączone bezpośrednio (bez modemów), sprzętowe sterowanie przepływem wymaga skrzyżowania linii RTS i CTS.

W przypadku bezpośredniego połączenia, terminal wysyłający dane musi przełączyć swoją linię CTS na stan ON (poprzez połączenie własnych linii RTS i CTS). W innym przypadku nadajnik nie będzie nadawać.

Nadajniko-odbiorniki 8250/16450/16550 wykorzystywane w komputerach IBM PC nie przetwarzają sygnału CTS sprzętowo, ale odzwierciedlają jego stan w rejestrze MSR (patrz punkt 2.5). Implementacja protokołu RTS/CTS jest zadaniem usługi BIOS Int14H. Nazwanie tej usługi mechanizmem sprzętowym nie jest całkowicie poprawne. Jeśli program wykorzystujący port COM komunikuje się z układem UART na poziomie rejestrów (a nie na poziomie BIOSu), wówczas w celu obsługi protokołu sam obsługuje przetwarzanie sygnału CTS. Niektóre programy komunikacyjne umożliwiają ignorowanie sygnału CTS (jeśli modem nie jest używany) i nie wymagają połączenia wejścia CTS z wyjściem własnego sygnału RTS. Są jednak różne nadajniko-odbiorniki (na przykład 8251), w którym sygnał CTS jest przetwarzany przez mechanizmy sprzętowe. W ich przypadku, a także dla „uczciwych" programów, wykorzystanie sygnału CTS w złączach (a nawet w kablach) jest obowiązkowe.

Programowe sterowanie przepływem XON/XOFF zakłada istnienie dwukierunkowego kanału transmisji danych. Protokół działa w następujący sposób: jeżeli urządzenie odbierające odkryje powody, dla których nie może kontynuować odbierania danych, wysyła w szeregowym kanale zwrotnym sygnał XOFF (13h). Po odebraniu tego sygnału urządzenie znajdujące się po przeciwnej stronie zawiesza transmisję. Kiedy urządzenie odbiorcze ponownie uzyskuje gotowość do odbierania danych, wysyła znak XON (11h). Urządzenie po drugiej stronie łącza po odebraniu tego sygnału wznawia transmisję. W porównaniu ze sprzętową implementacją protokołu, czas odpowiedzi nadajnika na zmianę stanu odbiornika zwiększa się co najmniej o czas trwania transmisji sygnału (XON lub XOFF) plus czas, jaki zajmuje programowi nadajnika reakcja na odebranie znaku (rysunek 2.14). Z tego powodu tylko odbiornik wyposażony w dodatkowy bufor dla odebranych danych informujący z góry o braku możliwości odbierania danych (w czasie, kiedy posiada jeszcze miejsce w buforze) może działać bez strat danych.



Rysunek 2.14. *Programowe sterowanie przepływem XON/XOFF*

Zaletą programowego sterowania przepływem jest brak konieczności transmisji sygnałów sterujących interfejsu: najprostszy kabel umożliwiający dwukierunkową wymianę składa się z zaledwie trzech żył (patrz rysunek 2.5a). Wada polega na tym, że oprócz konieczności korzystania z bufora oraz długiego czasu odpowiedzi (który obniża całkowitą wydajność kanału z powodu oczekiwania na sygnał XON), zaimplementowanie pełnodupleksowego trybu wymiany jest trudne. W tym przypadku istnieje konieczność uzyskania sygnałów sterowania przepływem ze strumienia danych odebranych oraz ich przetworzenia, co ogranicza liczbę znaków, które można przesłać.

Oprócz tych dwóch powszechnie używanych standardowych protokołów, obsługiwanych przez systemy operacyjne oraz większość urządzeń peryferyjnych, istnieją protokoły użytkownika wymagające specjalnej obsługi.

2.5. Układy scalone asynchronicznych nadajniko-odbiorników (UART)

W portach COM konwersja danych równoległych na szeregowe w celu realizacji transmisji danych oraz odwrotna konwersja w celu ich odbioru jest wykonywana przez dedykowane układy UART (ang. *Universal Asynchronous Receiver-Transmitter* — Uniwersalny asynchroniczny odbiorniko-nadajnik). Te same układy scalone generują i przetwarzają sygnały sterujące interfejsu. Porty COM komputerów IBM PC XT/AT są oparte na układach scalonych zgodnych z układami UART serii i8250 na poziomie rejestrów: 8250/16450/16550A. Seria ta jest ulepszeniem modelu oryginalnego, została opracowana w celu zwiększenia szybkości działania i zmniejszenia zużycia energii, a także obciążenia procesora podczas intensywnej wymiany. Zwróćmy uwagę, że:

- Układy 8250 mają wadę projektową (występowanie fałszywych przerwań), co uwzględniono w systemie BIOS komputera XT.
- 8250A: poprawiono błędy poprzedniego modelu, ale w efekcie utracono zgodność z systemem XT BIOS. Takie układy scalone działają w niektórych modelach komputerów AT, ale nie można ich używać dla szybkości transmisji 9 600 b/s.
- 8250B: poprawiono błędy 8250 i 8250A, przywrócono zgodność z systemem XT BIOS poprzez odtworzenie oryginalnego błędu przerwania. Działa w komputerach AT w systemie DOS (oprócz szybkości 9 600 b/s).

Układy scalone 8250x charakteryzują się niższymi szybkościami działania jeśli chodzi o żądania ze strony magistrali systemowej. W przypadku tych układów można odwoływać się do ich rejestrów w kolejnych cyklach magistrali procesora. Dla zapewnienia prawidłowej pracy, pomiędzy wywołaniami procesora muszą być wstawione opóźnienia programowe.

W komputerach klasy AT wprowadzono następujące modyfikacje układów scalonych UART:

- 16450: szybka wersja układu 8250 przeznaczona dla komputerów AT. Nie zawiera błędów układu 8250 ani pełnej zgodności z systemem XT BIOS.
- 16550: rozwinięta wersja układu 16450. Może wykorzystywać kanał DMA do wymiany danych. Jest także wyposażony w bufor FIFO, z którego nie można korzystać z powodu wadliwego działania.
- 16550A: układy charakteryzujące się szesnastobajtowymi wejściowymi i wyjściowymi buforami FIFO oraz wykorzystaniem kanałów DMA. Układy te są najczęściej używane w komputerach klasy AT dla transmisji z szybkościami przekraczającymi 9 600 b/s. Większość kontrolerów wejścia-wyjścia nowoczesnych chipsetów zapewnia zgodność z tym układem.

Z punktu widzenia programowania, układy scalone UART 16550A UART to zbiór rejestrów, do których dostęp jest określony przez adres (przesunięcie rejestru adresu w stosunku do bazowego adresu portu) oraz wartość bitu DLAB (7 bit rejestru LCR). W przestrzeni adresowej wejścia-wyjścia układ scalony zajmuje osiem kolejnych adresów. W tabeli 2.4 wyszczególniono rejestry układu UART 16550A wraz z metodami dostępu. Układy 8250 różnią się od nich tym, że nie zawierają rejestru FCR i wszystkich możliwości obsługi buforów FIFO i kanału DMA.

Tabela 2.4. *Rejestry układów UART 16550A*

Dostęp		Rejestr		Odczyt/Zapis
Przesunięcie	**DLAB**	**Nazwa**	**Funkcja**	
0h	0	THR	Rejestr transmisji.	Z
0h	0	RBR	Rejestr bufora odbiornika.	O
0h	1	DLL	Mniej znaczący bajt dzielnika szybkości transmisji.	O/Z
1h	1	DLM	Bardziej znaczący bajt dzielnika szybkości transmisji.	O/Z
1h	0	IER	Rejestr zezwolenia na przerwanie.	O/Z
2h	x	IIR	Rejestr identyfikacji przerwania.	O
2h	x	FCR	Rejestr sterujący buforami FIFO.	Z
3h	x	LCR	Rejestr sterujący łącza.	O/Z
4h	x	MCR	Rejestr sterujący modemu.	O/Z
5h	x	LSR	Rejestr stanu łącza.	O/Z[1]
6h	x	MSR	Rejestr stanu modemu.	O/Z[1]
7h	x	SCR	Rejestr roboczy.	O/Z

[1] *Niektóre bity są tylko do odczytu. Zapis do rejestru może doprowadzić do awarii protokołu.*

THR: rejestr transmisji (ang. *Transmitter Holding Register*) — tylko zapis. Dane zapisane do rejestru są wysyłane do wyjściowego rejestru transmisji (kiedy stanie się dostępny), stamtąd są umieszczane w wyjściowym sygnale TXD. Bit 0. (najmniej znaczący) jest wysyłany (i odbierany) jako pierwszy. Jeśli pakiet jest krótszy niż osiem bitów, bity najbardziej znaczące są ignorowane.

RBR: rejestr bufora odbiornika (ang. *Receiver Buffer Register*) — tylko odczyt. Dane szeregowe (RXD) odbierane w rejestrze przesunięcia odbiornika są umieszczane w rejestrze RBR. Stamtąd mogą być odczytane przez procesor. Jeśli bajt nie zostanie przeczytany do czasu odebrania następnego bajtu, występuje błąd przepełnienia. Jeśli pakiet jest krótszy niż osiem bitów, bity najbardziej znaczące mają wartość zero.

DLL: rejestr mniej znaczącego bajtu dzielnika prędkości przesyłania danych.

DLM: rejestr bardziej znaczącego bajtu dzielnika prędkości przesyłania danych. Dzielnik jest obliczany ze wzoru D = 115200/V, gdzie V oznacza szybkość transmisji w b/s. Wejściową częstotliwość synchronizacji 1,8432 MHz dzieli się przez uzyskany współczynnik, co daje częstotliwość transmisji danych wykorzystywaną przez układ UART.

IER: rejestr zezwolenia na przerwanie (ang. *Interrupt Enable Register*). Wartości bitów równe logicznej jedynce zezwalają na przerwanie z odpowiedniego źródła danych.

Funkcje bitów rejestru IER:

- bity[7:4]=0: nieużywane;
- bit 3. — Mod_IE: zmiana stanu modemu (dowolnej linii CTS, DSR, RI lub DCD);
- bit 2. — RxL_IE: przerwa łącza lub jego błąd;
- bit 1. — TXD_IE: zakończenie transmisji;
- bit 0. — RXD_IE: odebranie znaku (przekroczenie limitu czasu w trybie FIFO).

IIR: rejestr identyfikacji przerwania i wskaźnik stanu bufora FIFO (tylko do odczytu). W celu uproszczenia analizy programowej, w układzie UART zorganizowano wewnętrzne przerwania w system z czterema priorytetami. Kolejność priorytetów (w porządku rosnącym) to: stan łącza, odbiór znaku, wypróżnienie rejestru nadajnika, stan modemu. Kiedy wystąpi warunek przerwania, układ UART wskazuje na źródło o najwyższym priorytecie do czasu obsłużenia sytuacji przez wykonanie odpowiednich działań. Dopiero wtedy możliwa jest obsługa żądania z kolejnego źródła. Funkcje rejestru IIR opisano poniżej.

- bity [7:6] — stan trybu FIFO:
  - 11: tryb FIFO układu 16550A;
  - 10: tryb FIFO układu 16550;
  - 00: tryb standardowy;
- bity [5:4]: nieużywane;
- bit 3.: przerwanie przekroczenia limitu czasu odbioru w trybie FIFO (w buforze są dane do odczytania);

- bity [2:1]: źródło przerwania o najwyższym priorytecie (w trybie standardowym — bez użycia buforów FIFO):
  - 11: błąd łącza (przerwa). Wartość ustawiana poprzez odczyt rejestru stanu łącza;
  - 10: odebrano znak. Wartość ustawiana poprzez odczyt danych;
  - 01: wysłano znak (rejestr THR jest pusty). Wartość ustawiana poprzez zapis danych;
  - 00: zmienił się stan modemu. Wartość ustawiana poprzez odczyt rejestru stanu modemu;
- Bit 0: wskaźnik zaległego żądania przerwania (1: brak zaległego przerwania, 0: zaległe przerwanie).

W trybie FIFO źródło przerwania jest identyfikowane przez bity [3:1].

- 011: błąd łącza (przerwa). Wartość ustawiana poprzez odczyt rejestru stanu łącza;
- 010: odebrano znak. Wartość ustawiana poprzez odczyt rejestru danych odbiornika;
- 110: wskaźnik przekroczenia limitu czasu (w czasie równym czterokrotnemu okresowi potrzebnemu na wysłanie lub odebranie znaku nie został wysłany, ani odebrany znak, pomimo tego, że w buforze znajduje się co najmniej jeden). Wartość ustawiana poprzez odczyt rejestru danych odbiornika;
- 001: rejestr THR jest pusty. Wartość ustawiana poprzez zapis danych;
- 000: zmiana stanu modemu (CTS, DSR, RI lub DCD). Wartość ustawiana poprzez odczyt rejestru stanu modemu (MSR).

FCR: rejestr sterujący buforami FIFO(ang. *FIFO Control Register*) — tylko odczyt. Funkcje jego bitów opisano poniżej:

- bity [7:6] ITL (ang. *Interrupt Trigger Level* — poziom generowania przerwania): poziom zapełnienia bufora FIFO powodujący wygenerowanie przerwania:
  - 00: 1 bajt (domyślnie);
  - 01: 4 bajty;
  - 10: 8 bajtów;
  - 11: 14 bajtów;
- bity [5:4] — zarezerwowane.
- bit 3. — wybór trybu działania DMA.
- bit 2. — RESETTF (ang. *Reset Transmitter FIFO*): zerowanie licznika bufora FIFO nadajnika poprzez zapisanie logicznej jedynki; rejestr przesunięcia nie jest zerowany).
- bit 1. — RESETTRF (ang. *Reset Receiver FIFO*): zerowanie licznika bufora FIFO odbiornika poprzez zapisanie logicznej jedynki; rejestr przesunięcia nie jest zerowany).

- bit 0. — TRFIFOE (ang. *Transmit and Receive FIFO Enable*): włączenie (poprzez ustawienie logicznej jedynki) buforów FIFO nadajnika i odbiornika. W momencie zmiany stanu bufory są zerowane automatycznie.

LCR: rejestr sterowania łączem (ang. *Line Control Register*) — ustawianie parametrów kanału. Funkcje bitów rejestru LCR opisano poniżej.

- bit 7. — DLAB (ang. *Divisor Latch Access Bit*): kontrola dostępu do dzielnika szybkości przesyłania danych.
- bit 6. — BRCON (ang. *Break Control*):generowanie przerwy w łączu (wysyłanie zer) w przypadku, kiedy bit BRCON = 1;
- bit 5. — STICPAR (ang. *Sticky Parity*): wymuszone generowanie bitu parzystości:
  - 0: bit kontroli parzystości jest generowany zgodnie z parzystością znaku wychodzącego;
  - 1: stała wartość bitu kontroli parzystości: logiczne zero, jeśli EVENPAR = 1 oraz logiczna jedynka, jeśli EVENPAR = 0.
- bit 4. — EVENPAR (ang. *Even Parity select*): wybór typu kontroli parzystości: zero — nieparzystość, jeden — parzystość;
- bit 3. — PAREN (ang. *Parity Enable*): włączenie bitu kontroli parzystości:
  - 1: bit kontroli parzystości jest włączony (zgodny z bitem parzystości danych lub stały);
  - 0: bit kontroli parzystości jest wyłączony;
- bit 2. — STOPB (ang. *Stop Bits*): liczba bitów stopu:
  - 0: 1 bit stopu;
  - 1: 2 bity stopu (w przypadku kodu pięciobitowego, bit stopu ma długość 1,5 bitu);
- bity [1:0] — SERIALDB (ang. *Serial Data Bits*): liczba bitów danych:
  - 00: 5 bitów;
  - 01: 6 bitów;
  - 10: 7 bitów;
  - 11: 8 bitów.

MCR: rejestr sterowania modemem (ang. *Modem Control Register*). Opis rejestru MCR zamieszczono poniżej.

- bity [7:5] =0: zarezerwowane;
- bit 4. — LME (ang. *Loopback Mode Enable*): włączenie trybu diagnostycznego:
  - 0: tryb normalnej transmisji;
  - 1: tryb diagnostyczny;

- bit 3. — IE (ang. *Interrupt Enable*): zezwolenie na przerwanie (sygnalizowane przez wyjście zewnętrzne OUT2):
  - 0: przerwania wyłączone;
  - 1: przerwania włączone;

W trybie diagnostycznym bit jest umieszczany w bicie rejestru wejściowego MSR.7 (emulacja DCD);

- bit 2. — OUT1C (ang. *OUT1 bit Control*): sterowanie sygnałem wyjściowym 1 (nieużywany); w trybie diagnostycznym bit jest umieszczany w bicie rejestru wejściowego MSR.6 (emulacja RI);
- bit 1. — RTSC (ang. *Request To Send Control*): sterowanie sygnałem wyjściowym RTS; w trybie diagnostycznym umieszczany w bicie rejestru wejściowego MSR.4 (emulacja CTS):
  - 0: aktywny (napięcie dodatnie);
  - 1: nieaktywny (napięcie ujemne);
- bit 0. — DTRC (ang. *Data Terminal Ready Control*): sterowanie sygnałem wyjściowym DTR. W trybie diagnostycznym umieszczany w bicie rejestru wejściowego MSR.5 (emulacja DSR):
  - 0: aktywny (napięcie dodatnie);
  - 1: nieaktywny (napięcie ujemne).

LSR: rejestr stanu łącza (ang. *Line Status Register*) — stan nadajniko-odbiornika. Funkcje rejestru LSR opisano poniżej.

- bit 7. — FIFOE (ang. *FIFO Error status*): błąd danych w trybie FIFO (w buforze znajduje się co najmniej jeden znak, dla którego wystąpił błąd formatu, parzystości lub przerwa); w trybie bez wykorzystania buforów FIFO bit jest zawsze ustawiany na wartość zero;
- bit 6. — TEMPT (ang. *Transmitter Empty Status*): rejestr nadajnika jest pusty (w rejestrze przesunięcia, rejestrze THR lub w buforze FIFO nie ma danych do wysłania);
- bit 5. — THRE (ang. *Transmitter Holding Register Empty*): rejestr nadajnika jest gotowy do odebrania bajtu przeznaczonego do transmisji; w trybie FIFO oznacza to, że bufor FIFO transmisji jest pusty; może być źródłem przerwania;
- bit 4. — BD (ang. *Break Detected*): wskaźnik przerwy w łączu (w rejestrze wejściowym odbiornika jest wartość zero przez okres nie krótszy niż trwa operacja wysłania znaku);
- bit 3. — FE (ang. *Framing Error*): błąd ramki (niepoprawny bit stopu);
- bit 2. — PE (ang. *Parity Error*): błąd parzystości;
- bit 1. — OE (ang. *Overrun Error*): przepełnienie (utrata znaku); jeśli odbiór następnego znaku rozpocznie się przed przesłaniem poprzedniego z rejestru przesunięcia do bufora lub rejestru FIFO, pierwszy znak jest tracony;

- bit 0. — DR (ang. *Receiver Data Ready*): odebrane dane są gotowe (w rejestrze DHR lub buforze FIFO); wartość ustawiana poprzez odczyt rejestru danych odbiornika.

Wskaźniki błędów — bity [4:1] — są zerowane po odczytaniu rejestru LSR. W trybie FIFO wskaźniki błędów są zapisane w buforze FIFO razem z przesyłanymi znakami. Są umieszczane w rejestrze (i generują przerwania) w momencie, kiedy znak odebrany z błędami znajduje się na szczycie bufora FIFO (w wierszu do odczytania). W przypadku przerwy w łączu, tylko znak „przerwy" jest umieszczany w buforze FIFO, po czym układ UART czeka na odtworzenie łącza i następny bit startu.

MSR: rejestr stanu modemu (ang. *Modem Status Register*). Funkcje rejestru MSR opisano poniżej.

- bit 7. — DCD (ang. *Data Carrier Detect* — wykryto nośną danych): stan linii DCD:
  - 0: aktywny (napięcie dodatnie);
  - 1: nieaktywny (napięcie ujemne);
- bit 6. — RI (ang. *Ring Indicator* — Wskaźnik dzwonka): stan linii RI:
  - 0: aktywny (napięcie dodatnie);
  - 1: nieaktywny (napięcie ujemne);
- bit 5. — DSR (ang. *Data Set Ready* — zbiór danych gotowy): stan linii DSR:
  - 0: aktywny (napięcie dodatnie);
  - 1: nieaktywny (napięcie ujemne);
- bit 4. — CTS (ang. *Clear To Send* — zezwolenie na wysyłanie): stan linii CTS:
  - 0: aktywny (napięcie dodatnie);
  - 1: nieaktywny (napięcie ujemne);
- bit 3. — DDCD (ang. *Delta Data Carrier Detect*): zmiana stanu linii DCD;
- bit 2. — TERI (ang. *Trailing Edge Of Ring Indicator* — koniec sygnału dzwonienia): zbocze opadające sygnału RI (koniec dzwonienia);
- bit 1. — DDSR (ang. *Delta Data Set Ready*): zmiana stanu linii DSR;
- bit 0. — DCTS (ang. *Delta Clear To Send*): zmiana stanu sygnału CTS.

Wskaźniki zmian stanu (bity [3:0] są ustawiane poprzez odczyt rejestru.

SCR: (ang. *Scratch Register* — rejestr roboczy) — rejestr roboczy o szerokości 8 bitów. Nie ma wpływu na działanie układu UART i służy do przechowywania danych tymczasowych (w układach 8250 nie ma tego rejestru).

W trybie diagnostycznym (jeśli LME =1) wewnątrz układu UART tworzona jest wewnętrzna pętla:

- Wyjścia nadajnika są przełączane w stan logicznej jedynki.
- Wejście odbiornika jest rozłączane.

- Wyjście rejestru przesunięcia nadajnika jest logicznie łączone z wejściem odbiornika.
- Linie DSR, CTS, RI i DCD są rozłączane od linii wejściowych i są kontrolowane wewnętrznie przez bity DTRC, RTSC, OUTIC i IE.
- Wyjścia sygnałów sterowania modemem są przełączane w stan pasywny (logiczne zero).

Dane przesyłane szeregowo są odbierane natychmiast, co umożliwia diagnozowanie wewnętrznego kanału danych portu (włącznie z rejestrami przesunięć), obsługę przerwań, a także określenie szybkości działania układu UART.

2.6. Obsługa portów COM w systemie operacyjnym

Porty COM są obsługiwane przez usługę BIOS int14h, która zapewnia wykonywanie następujących funkcji:

- *Inicjalizacja* (ustawienie szybkości wymiany i formatu transmisji; wyłączenie źródeł przerwań). Nie ma wpływu na sygnały DTR i RTS (po zresetowaniu sprzętu sygnały są nieaktywne).
- *Wysyłanie znaku*. Uaktywnienie sygnałów DTR i RTS. Po zwolnieniu rejestru THR jest do niego wprowadzany wysyłany znak.
- *Odbiór znaku*. Uaktywnienie tylko sygnału DTR (sygnał RTS przechodzi w stan nieaktywny). Sterownik oczekuje na odbiór znaku.
- *Odpytywanie stanu* modemu i łącza (odczytanie rejestrów MSR i LSR).

Przerwania sprzętowe nie są używane. Oczekiwanie na gotowość do wejścia i wyjścia jest ograniczona limitem czasu. Gotowość można natychmiast sprawdzić poprzez odpytywanie stanu, przed próbą odebrania lub wysłania znaku.

Podczas wykonywania procedury POST system BIOS sprawdza obecność portu szeregowego (rejestrów układu UART 8250 lub zgodnych) pod standardowymi adresami i umieszcza bazowe adresy wykrytych portów w komórkach obszaru danych BIOSu — BIOS Data Area 0:0400h, 0:0402h, 0:0404h i 0:0406h. W komórkach są zapisywane adresy portów o logicznych nazwach od COM1 do COM4. Zerowa wartość adresu oznacza, że port o określonym numerze nie istnieje. Ustawienia limitów czasu dla portów są zapisywane w komórkach 0:47Ch, 0:47Dh, 0:473Eh oraz 0:47Fh.

Wykryte porty są inicjowane na szybkość wymiany 2 400 b/s, siedem bitów danych, parzystość i jeden bit stopu. Sygnały sterowania interfejsem DTR i RTS są przełączane w stan początkowy (OFF: ujemny poziom linii).

2.7. Konfiguracja portów COM

W komputerze mogą być zainstalowane maksymalnie cztery porty szeregowe od COM1 do COM4. W maszynach klasy AT zazwyczaj występują dwa porty. Proces zarządzanie portów szeregowych jest podzielony na dwie fazy: konfiguracja wstępna sprzętu portów oraz zmiany trybów roboczych poprzez oprogramowanie aplikacyjne lub systemowe. Konfiguracja portów COM zależy od sposobu implementacji portów. Porty fizycznie umieszczone na kartach rozszerzeń konfiguruje się za pomocą zworek umieszczonych na samych kartach. Porty wbudowane na płytach głównych konfiguruje się w systemie BIOS.

Można skonfigurować następujące parametry:

- *Adres bazowy*. Porty COM1– COM4 najczęściej mają adresy 3F8h, 2F8h, 3E8h i 2E8h. Podczas inicjalizacji BIOS sprawdza pod tymi adresami obecność portów dokładnie w tej kolejności i przypisuje im logiczne nazwy COM1, COM2, COM3 i COM4. Porty COM3 i COM4 mogą mieć alternatywne adresy, odpowiednio 3E0h, 338h oraz 2E0h, 238h. Dla komputerów PS/2, standardowe adresy od COM3 do COM8 to odpowiednio 3220h, 3228h, 4220h, 4228h, 5220h oraz 5228h.
- *Wykorzystywane linie żądania przerwań*: dla portów COM1 i COM3 zwykle używane są linie `IRQ4` lub `IRQ11`, dla portów COM2 i COM4: linie `IRQ3` lub `IRQ10`. Z technicznego punktu widzenia, numery przerwań można przypisywać do adresów bazowych (numerów portów) w dowolnej kolejności, ale niektóre programy i sterowniki (na przykład sterowniki myszy szeregowej) są zaprogramowane do wykorzystania standardowych kombinacji. Do każdego portu, który potrzebuje sprzętowego przerwania, jest przypisywana oddzielna linia, która nie może kolidować z liniami żądania przerwań używanymi przez inne urządzenia. Przerwania są wymagane tylko przez te porty, do których są podłączone urządzenia wejściowe, zasilacze UPS lub modemy. Przerwania dla transmisji do drukarek i ploterów są wykorzystywane tylko w wielozadaniowych systemach operacyjnych (choć nie zawsze). Dlatego do takich zastosowań można oszczędzić ten cenny zasób. Przerwań nie używa się także do połączenia dwóch komputerów za pomocą kabla modemu zerowego. To, czy linia żądania przerwań jest używana oddzielnie, czy współdzielona przez kilka urządzeń, zależy od sprzętowej implementacji połączenia oraz wykorzystywanego oprogramowania. W przypadku portów zainstalowanych na magistrali ISA, współdzielone przerwania zazwyczaj nie działają.
- *Kanał DMA* (dla układów scalonych UART 16450/16550 umieszczonych na płycie głównej): włączenie kanału DMA i podanie numeru kanału. Dla portów COM rzadko używa się kanału DMA.

2.8. Zastosowania portów COM

Porty COM powszechnie wykorzystuje się do podłączania różnego rodzaju urządzeń peryferyjnych i sprzętu komunikacyjnego do nawiązywania połączeń ze sprzętem technicznym, na przykład sterującym i monitorującym, programatorami, emulatorami układów i innymi urządzeniami wykorzystującymi protokół RS-232C.

Najczęściej porty COM są wykorzystywane do podłączania urządzeń wskaźnikowych (myszy, trackballi). W takim przypadku porty są wykorzystywane w trybie wprowadzania szeregowych danych. Mysz z interfejsem szeregowym można podłączyć do dowolnego działającego portu. W celu dopasowania złączy portu i myszy, czasami wykorzystuje się przejściówki DB-9S — DB25P lub DB-25S — DB-9P. Myszy do prawidłowego działania wymagają przerwań IRQ4 dla portu COM1 oraz IRQ3 dla portu COM2. Cechą szczególną sterownika jest fakt, że w celu używania myszy port COM1 musi korzystać z przerwania IRQ4. Dla użytkownika ważne jest, aby zdać sobie sprawę z samego ograniczenia. Każde zdarzenie: ruch myszą lub wciśnięcie (zwolnienie) przycisku jest kodowane binarnie i wysyłane poprzez interfejs RS-232C. W tym celu wykorzystuje się transmisję asynchroniczną: źródłem bipolarnego zasilania są linie sterujące interfejsu.

W celu podłączenia *modemów zewnętrznych* wykorzystywany jest kabel DTE–DCE z kompletem (dziewięciu) połączeń. Schemat połączeń takiego kabla pokazano na rysunku 2.4. Ten sam typ kabla wykorzystuje się w celu dopasowania złączy (według liczby styków). W tym celu można także wykorzystać przejściówki myszy 9 na 25. Oprogramowanie komunikacyjne zazwyczaj korzysta z przerwań, ale w tym przypadku istnieje swoboda wyboru numeru portu (adresu) i linii przerwania. Jeśli planuje się operacje z szybkością 9 600 b/s lub wyższą, port COM należy zaimplementować za pomocą układu UART 16550A lub zgodnego. Zakres, w jakim są dostępne bufory FIFO oraz własności kanału DMA, zależy od wykorzystywanego oprogramowania komunikacyjnego.

Porty COM można także wykorzystać w celu połączenia dwóch komputerów znajdujących się w niewielkiej odległości od siebie za pomocą kabla zerowego modemu (patrz rysunek 2.5). Programy takie jak Norton Commander lub MS-DOS Interlink umożliwiają wymianę plików z szybkością do 115.2 kb/s bez wykorzystania przerwań sprzętowych. Takie samo połączenie można także wykorzystać przez pakiet sieciowy Lantastic oferujący bogatsze własności oraz przez usługi systemu operacyjnego Windows.

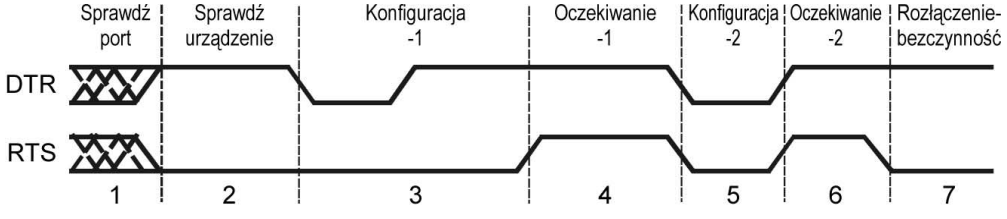
Porty COM umożliwiają podłączanie *kluczy elektronicznych* (urządzeń zabezpieczających) mających na celu zabezpieczenia przed nieuprawnionym wykorzystywaniem oprogramowania. Urządzenia te mogą być „przezroczyste" (zapewniające możliwość wykorzystania portu do podłączenia innego sprzętu) lub całkowicie przejmować port.

Przy pomocy odpowiedniego oprogramowania port COM umożliwia przekształcenie komputera PC w terminal emulujący zestaw poleceń popularnych specjalizowanych terminali (VT-52, VT-100, itp.). Najprostszy terminal można uzyskać poprzez połączenie funkcji BIOS obsługi portu COM (int14h), zdalnego wyprowadzania danych (int10h) oraz wprowadzania danych z klawiatury (Int16h). Taki terminal może jednak działać tylko z niskimi szybkościami transmisji, ponieważ chociaż funkcje BIOSu są uniwersalne, nie są zbyt szybkie.

Porty COM można także wykorzystać jako dwukierunkowy interfejs z trzema sterowanymi programowo liniami wyjściowymi (TD, DTR i RTS) oraz czterema liniami wejściowymi (CTS, DSR, DCD i RI) wykorzystującymi sygnały bipolarne. Można je wykorzystać na przykład do programowej implementacji synchronicznych interfejsów szeregowych (patrz podrozdział 10.5). W czasach komputerów AT-286 wykorzystywano jednobitowe konwertery impulsowe umożliwiające zapisywanie sygnałów audio w komputerze PC za pomocą linii wejściowej portu COM. Odtwarzanie tak zapisanego dźwięku w głośniku komputera PC umożliwiało uzyskanie na przykład sygnałów mowy.

2.9. Porty COM i PnP

Nowoczesne urządzenia peryferyjne podłączone do portu COM obsługują specyfikację PnP. Główne zadanie systemu operacyjnego polega na zidentyfikowaniu podłączonego urządzenia. Do tego celu opracowano prosty protokół, który można zaimplementować dla każdego portu COM za pomocą środków wyłącznie programowych. Mechanizm działania tego protokołu pokazano na rysunku 2.15.



Rysunek 2.15. *Żądanie identyfikatora urządzenia PnP*

1. *Sprawdź port*: sprawdzenie dostępności portu. Jeśli jest wolny, jest inicjowany do stanu bezczynności ze stanami linii `DTR=ON`, `RTS=OFF`, `TXD=Mark`. Jeśli jest zajęty, dalsze etapy protokołu nie są wykonywane.
2. *Sprawdź urządzenie*: sprawdzenie obecności urządzenia. Przez pewien okres czasu (0,2 s) system oczekuje na sygnał `DSR`, co oznacza, że do portu jest podłączone urządzenie. W najprostszym przypadku sygnały `DTR-DSR` w urządzeniu są zwarte, co umożliwia uzyskanie odpowiedzi. Po wykryciu urządzenia wykonywane są kolejne etapy protokołu (w celu odebrania szeregowych danych z urządzenia wykorzystywane są sygnały sterowania liniami `DTR` i `RTS`). Jeśli nie ma odpowiedzi, protokół przełącza się do bezczynnego stanu rozłączenia. W takim przypadku systemy operacyjne obsługujące dynamiczną konfigurację okresowo wykonują poprzednią sekwencję po to, aby sprawdzić, czy nie podłączono urządzenia.
3. *Konfiguracja-1*. Port jest programowany na szybkość 1 200 b/s, siedem bitów danych, brak kontroli parzystości, jeden bit stopu. Na linii `DTR` jest stan niski przez 0,2 s. Następnie sygnał `DTR` jest ustawiany na logiczną jedynkę, a po kolejnym okresie 0,2 s sygnał `RTS` także jest ustawiany na logiczną jedynkę.
4. *Oczekiwanie-1*. System operacyjny oczekuje na pierwszy znak z urządzenia przez okres 0,2 sekundy. Kiedy znak nadejdzie, system rozpoczyna odbiór identyfikatora. Jeśli podczas tego czasu znak nie nadejdzie, system operacyjny wykonuje drugą próbę konfiguracji (patrz punkt 5), która nieco różni się od pierwszej.
5. *Konfiguracja-2*. Sygnały `DTR` i `RTS` są ustawiane na logiczne zero na okres 0,2 sekundy. Po tym czasie oba sygnały są przełączane na logiczną jedynkę.
6. *Oczekiwanie-2*. System operacyjny oczekuje na pierwszy znak z urządzenia przez okres 0,2 sekundy, a kiedy ten nadejdzie, zaczyna odbierać identyfikator. Jeśli w tym czasie znak nie nadejdzie, w zależności od stanu sygnału `DSR` wprowadzany jest stan *weryfikacja rozłączenia* (jeśli `DSR=0`) lub *połączenie-bezczynność* (jeśli `DSR=1`).

7. W stanie *Połączenie-bezczynność* sygnał DTR jest ustawiany na logiczną jedynkę, a sygnał RTS na logiczne zero. Port jest programowany na szybkość 300 b/s, siedem bitów danych, brak kontroli parzystości i jeden bit stopu. Jeśli w czasie trwania tego stanu zostanie wykryty stan DSR=0, system operacyjny powinien poinformować, że urządzenie zostało rozłączone.

Odbiór identyfikatora urządzenia po jednym znaku naraz wiąże się z ograniczeniami czasowymi 0,2 s na każdy znak, a także całkowitym limitem czasu 2,2 s, co umożliwia odbieranie ciągów o długości do 256 znaków. Ciąg znaków identyfikatora PnP musi zawierać znaczniki początku i końca (odpowiednio 28h lub 08h oraz 29h lub 09h). Pomiędzy nimi umieszczana jest treść identyfikatora w standardowym formacie. Przed znacznikiem startu można umieścić ciąg szesnastu znaków niezwiązanych z identyfikatorem PnP. Jeśli podczas pierwszych 0,2 s oczekiwania na znak (krok 4. lub 6.) nie nadejdzie znacznik startu lub zostanie przekroczony limit czasu i nie nadejdzie znacznik końca albo znaki zostaną odebrane z błędami, nastąpi przełączenie do stanu *połączenie-bezczynność*. W przypadku odebrania poprawnego ciągu identyfikatora, jest on przekazywany do systemu operacyjnego.

W celu wprowadzenia stanu *weryfikacja rozłączenia* sygnał DTR jest ustawiany na logiczną jedynkę, sygnał RTS na logiczne zero i co pięć sekund sprawdzany jest stan sygnału DSR. Jeśli DSR=1, następuje przełączenie do stanu *połączenie-bezczynność* (zobacz krok 7.), a jeśli DSR=0, do stanu *rozłączenie-bezczynność*, w którym system okresowo odpytuje sygnał DSR w celu sprawdzenia, czy podłączono urządzenie.

Opisany mechanizm został opracowany przez firmę Microsoft z uwzględnieniem zgodności z urządzeniami, które nie są typu PnP. W ten sposób mamy pewność, że urządzenia te nie będą wyłączone oraz system zachowa stabilność w przypadku odbioru komunikatów nie będących identyfikatorami PnP. Na przykład konwencjonalna mysz Microsoft Mouse po otrzymaniu zasilania z interfejsu odpowiada znakiem „M” (mysz wyposażona w trzy przyciski ciągiem „M3”).

2.10. Problemy z działaniem portów COM i ich rozwiązywanie

Nieprawidłowe działanie portów COM często występuje podczas instalacji nowych portów lub niepomyślnego podłączania urządzeń zewnętrznych.

2.10.1. Testowanie konfiguracji

Rozwiązywanie problemów z portami szeregowymi (podobnie jak z równoległymi) rozpoczyna się od sprawdzenia, czy system je identyfikuje. Lista adresów zainstalowanych portów zazwyczaj pojawia się w tabelce, którą system BIOS wyświetla przed załadowaniem systemu operacyjnego. Listę tę można również przeglądać za pomocą programów diagnostycznych lub bezpośrednio w obszarze danych BIOSu za pomocą debuggera.

Jeśli system BIOS wykryje mniej portów, niż fizycznie zainstalowano, prawdopodobnie dwóm portom przypisano ten sam adres lub jednemu z nich przypisano nieprawidłowy adres. Problemy mogą pojawić się w przypadku adresów portów COM3 i COM4: nie wszystkie wersje BIOSu będą poszukiwać portów pod alternatywnymi adresami 3E0h, 338h, 2E0h i 238h. Czasami wyszukiwanie pod adresami 3E8h i 2E8h nie jest wykonywane. Sposób wyświetlania zidentyfikowanych portów na liście może być mylący: w przypadku zainstalowania dwóch portów o adresach 3F8h i 3E8h, na liście będą one występowały jako COM1 i COM2 i za pomocą tych nazw będzie można się do nich odwoływać. Te same porty mogą jednak pojawić się na liście jako COM1 i COM3 (ponieważ 3E8h to standardowy adres portu COM3). W tym przypadku odwołanie do portu COM3 nie powiedzie się, ponieważ adres 3E8h będzie umieszczony w komórce 0:402h obszaru danych BIOSu, która odpowiada portowi COM2, a w komórce pamięci portu COM3 (0:404h) będzie wartość zero: wskaźnik, że nie ma takiego portu. Użytkownik może „wyjaśnić" systemowi który port jest który ręcznie, za pomocą dowolnego debuggera, poprzez wstawienie poprawnych adresów do komórek obszaru danych BIOSu (trzeba by to robić za każdym razem po ponownym załadowaniu systemu, a przed wykorzystaniem „zagubionego" portu). Odnajdywanie portów umożliwiają niektóre narzędzia diagnostyczne (np. *PortFinder*).

Jeśli dwóm portom przypisano ten sam adres, program diagnostyczny może wykryć błędy portów tylko wtedy, gdy jest wykorzystywany z zewnętrzną pętlą diagnostyczną. Programowe testowanie portu bez pętli nie spowoduje wykrycia błędów, ponieważ w tym przypadku nastąpi uaktywnienie trybu diagnostycznego (patrz opis układu UART) i porty wchodzące ze sobą w konflikt (choć indywidualnie działające poprawnie) będą działały równolegle, dzięki czemu informacje wynikowe będą poprawne. W normalnych warunkach pracy poprawne działanie portów wchodzących ze sobą w konflikt jest niemożliwe. Podczas rozwiązywania konfliktów adresów przydatne jest instalowanie portów po kolei i obserwowanie adresów pojawiających się na liście.

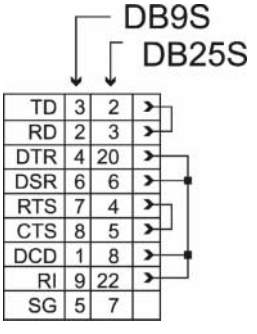
Jeśli fizycznie jest zainstalowany tylko jeden port, a system BIOS go nie znajdzie, powód może być ten sam co w przypadku portu LPT: albo go wyłączono podczas konfiguracji, albo nastąpiła awaria. Problemy z działaniem portu można rozwiązać poprzez poruszanie kartą adaptera w gnieździe płyty głównej.

Podczas korzystania z portów COM używane są *przerwania sprzętowe*: są one wykorzystywane przy podłączaniu modemów, myszy lub innych urządzeń wejściowych. Powodem nieprawidłowego działania tych urządzeń mogą być niepoprawne ustawienia linii żądania przerwań. Możliwe są konflikty z innymi urządzeniami, a także nieodpowiednie dobranie numerów przerwań z adresami portów.

2.10.2. Testowanie działania

Wstępne testowanie portu COM można wykonać za pomocą programów diagnostycznych (CheckIt) bez wykorzystania pętli. Takie testowanie umożliwia sprawdzenie układu scalonego UART (wewnętrzny tryb diagnostyczny) i generowanie przerwań, ale nie ma możliwości testowania układów buforów wejściowych i wyjściowych, które znacznie częściej są źródłem problemów. Jeśli test nie powiedzie się, przyczyn problemów należy szukać w konflikcie adresu (linii przerwań) lub w samym układzie UART.

Do dokładniejszego testowania zaleca się stosowanie zewnętrznych pętli podłączanych do złącza portu COM (rysunek 2.16). W odróżnieniu od portu LPT, port COM ma więcej linii wejściowych niż wyjściowych, co umożliwia dokładne testowanie wszystkich obwodów. Pętla łączy linie wejściowe odbiornika z liniami wyjściowymi nadajnika. Obowiązkowe dla wszystkich pętli połączenie RTS-CTS umożliwia działanie nadajnika: bez tego połączenia nie można przesyłać znaków. Do sprawdzania wejściowych sygnałów DSR, DCD i RI zazwyczaj wykorzystywany jest wyjściowy sygnał DTR.



Rysunek 2.16. *Pętla do sprawdzania portów COM (dla programów CheckIt i Norton Diagnostics)*

Jeśli test za pomocą zewnętrznej pętli nie powiedzie się, przyczyny problemów należy szukać w zewnętrznych buforach, ich zasilaczach lub przewodach taśmowych łączących adapter z zewnętrznymi złączami. Do diagnozowania tych problemów przyda się woltomierz lub oscyloskop. Procedura testowania może przebiegać w następujący sposób:

1. Sprawdź, czy zasilacze bipolarne obwodów wyjściowych nadajnika działają prawidłowo (ta czynność powinna być wykonana jako pierwsza, ale ponieważ z technicznego punktu widzenia jest najbardziej skomplikowana, można ją odłożyć na najgorszą ewentualność — sytuację, kiedy zaistnieje prawdopodobieństwo konieczności wymiany układów buforów).
2. Sprawdź napięcia na liniach wyjściowych TD, RTS i DTR: po wyłączeniu i ponownym włączeniu sprzętu na wyjściu TD powinien być potencjał ujemny o wartości około –12 V (zdecydowanie niższy niż – V). Podczas próby wysyłania znaku na liniach wyjściowych RTS i DTR powinno pojawić się napięcie dodatnie oraz seria impulsów bipolarnych na wyjściu TD. Jeśli nie ma tych sygnałów, istnieje możliwość, że popełniono błąd podczas podłączania płyty głównej do zewnętrznego złącza za pomocą przewodu taśmowego. Najbardziej prawdopodobne jest, że:
   - ♦ nie podłączono przewodu taśmowego;
   - ♦ przewód został podłączony nieprawidłowo (złącze jest odwrócone lub przesunięte);
   - ♦ przewód ma złącze, które nie pasuje do złącza na płycie głównej.

   Pierwsze dwa przypadki sprawdza się dokładnie oglądając połączenie, ale trzeci wymaga dodatkowych działań. W tabeli 2.1 zaprezentowano trzy znane autorowi układy przewodów dla dziesięciożyłowego przewodu taśmowego wykorzystywanego w portach COM. Być może dla portów wbudowanych na płytach głównych istnieją inne wersje. Teoretycznie istnieje możliwość zakupu przewodu taśmowego odpowiedniego dla danego złącza.

Jeśli źródłem problemu jest zły układ przewodu, trzy sygnały wyjściowe można zlokalizować na innych stykach złącza (napięcie na stykach wejściowych jest bardzo niskie). Jeśli nie uda się odnaleźć sygnałów wyjściowych na złączu, ewidentnie przyczyną problemów jest awaria obwodów buforów.

3. Po połączeniu linii RTS i CTS (lub wykorzystaniu pętli) należy spróbować wysłać do portu COM krótki plik (na przykład za pomocą polecenia COPY C:\AUTOEXEC.BAT COM1). Jeśli port działa prawidłowo, polecenie zostanie wykonane w kilka sekund, a po zakończeniu wyświetli się komunikat o sukcesie. W tym momencie napięcia na wyjściach RTS i DTR powinny zmienić się na ujemne, a na wyjściowej linii TD powinien pojawić się pakiet impulsów bipolarnych o amplitudzie większej niż 5 V. Jeśli napięcia na liniach RTS i DTR nie zmienią się, źródłem problemów są obwody buforów. Jeśli na wyjściu RTS (oraz na wejściu CTS) pojawi się napięcie ujemne, a polecenie COPY zakończy się błędem, najprawdopodobniej awarii uległ odbiornik linii CTS (albo jest problem z taśmą). Jeśli wykonanie polecenia COPY zakończyło się sukcesem, ale nie ma zmian na wyjściu TD (można je wykryć za pomocą woltomierza analogowego, ale nie można ocenić amplitudy), uszkodzeniu uległ nadajnik bufora sygnału TD.

Wymiana układów odbiornika i nadajnika jest znacznie prostsza, jeśli są one zamontowane w podstawkach. Przed wymianą układu scalonego należy się upewnić za pomocą oscyloskopu lub woltomierza, że element ten uległ uszkodzeniu.

Jeśli układy buforów są częścią układu scalonego LSI (co w dzisiejszych czasach jest dość częste), portu nie da się naprawić (przynajmniej nie za pomocą standardowych narzędzi). Można wyłączyć wadliwy port COM wbudowany na płycie głównej w programie BIOS Setup, ale w niektórych przypadkach może się zdarzyć, że port spali się razem z obwodem odpowiedzialnym za jego rozłączenie: w takiej sytuacji port będzie zajmował miejsce w przestrzeni adresów i blokował linię przerwań. Czasami uszkodzenie portu może spowodować uszkodzenie całej płyty głównej.

Przyczyną wadliwego działania portu mogą być złącza i kable. Złącza czasami nie gwarantują odpowiedniego styku, a kable, oprócz możliwych przerw, mogą mieć złą charakterystykę częstotliwościową. Jakość częstotliwości kabla zazwyczaj daje o sobie znać przy dużych szybkościach transmisji (56 kb/s lub 115 kb/s). Jeśli istnieje konieczność użycia długich kabli dla dużych szybkości, przewody sygnałowe powinny być ekranowane, a ekrany podłączone do oddzielnego przewodu masy.

2.10.3. Brak zasilania interfejsu, czyli dlaczego nie działa mysz

Urządzenia o niskim poborze mocy podłączane do portu COM są zasilane z linii wyjściowych interfejsu. Jeśli linie DTR lub TRS nie są wykorzystywane bezpośrednio w tym celu, do jakiego zostały przeznaczone, można je wykorzystać jako źródło zasilania zapewniającego napięcie o wartości około 12 V. Prąd zwarcia do obwodu masy jest ograniczony przez układ bufora nadajnika do wartości 20 mA. Podczas inicjalizacji portu, linie te są

przełączane w stan OFF, tzn. dostarczają napięcie ujemne. Na linii TD w stanie bezczynności jest logiczna jedynka, a zatem na wyjściu jest napięcie ujemne. Potencjały poszczególnych linii można kontrolować za pomocą rejestrów portu COM (na wyjściu TD pojawia się napięcie dodatnie w przypadku ustawienia bitu BRCON). Napięcie można także wyłączyć z linii sygnałowych poprzez diody stabilizacyjne wykorzystywane razem z kondensatorami. W tym przypadku należy wziąć pod uwagę czas, przez jaki sygnał wyjściowy pozostaje w pożądanym stanie (po to, aby zgromadzona energia była wystarczająca).

Wszystkie manipulatory (myszy) podłączone do portów COM pobierają zasilanie bipolarne z linii interfejsu (napięcie dodatnie z linii DTR lub RTS, napięcie ujemne z linii TD). W związku z tym, jeśli mysz podłączona do portu nie działa, powinniśmy sprawdzić napięcia na odpowiednich stykach złącza. Czasami zdarza się, że tylko konkretna mysz (model lub egzemplarz) nie działa z określonym portem, pomimo tego, że inne myszy działają w tym porcie bez problemu, a mysz, która nie chce działać, bez problemu działa w innych portach. W tym przypadku problemem mogą być poziomy napięć. Zgodnie ze standardem port nie powinien dostarczać napięcia niższego niż 5 V (wartość bezwzględna). Jeśli port zapewnia wyłącznie minimum, w przypadku niektórych myszy taki prąd nie wystarczy do zasilania diod LED (elementów zużywających najwięcej energii).

Port pobiera zasilanie bipolarne za pośrednictwem płyty głównej z zasilacza komputera. Jeśli na wyjściu zasilacza nie ma napięcia +12 V, zazwyczaj objawia się to brakiem działania dysków. Brak napięcia –12 V „zauważą" jedynie urządzenia podłączone do portu COM. Teoretycznie zasilacze monitorują napięcia na swoich wyjściach (informując o problemach za pomocą sygnału Power Good, który powoduje zresetowanie sprzętu). Czasami jednak są stosowane uproszczone zasilacze, które nie kontrolują wszystkich swoich napięć. Poza tym istnieje możliwość złego styku w złączu zasilającym na płycie głównej.