

Wykorzystanie portów komputera PC

Port szeregowy

1 część

W kilku ostatnich numerach EdW poświęcono dużo miejsca sprawie wykorzystania komputera PC do sterowania różnych urządzeń zewnętrznym.

Przedstawiono kilka książek, które w przystępny sposób uczą programowania w języku Basic. Napływające listy świadczą o znacznym zainteresowaniu tą sprawą. Jednak materiał zawarty w polecanych przez nas książkach nie obejmuje zagadnień ogromnie istotnych dla elektronika – sposobów wykorzystania portów komputera. Niniejszy cykl artykułów wypełnia tę lukę – zawiera obszerne i wyczerpujące omówienie zasady działania oraz możliwości dostępu do portów komputera PC. Przedstawiony materiał umożliwi nawet mało zaawansowanemu użytkownikowi komputera PC praktyczne wykorzystanie tych portów.

Oprócz informacji ogólnych Autor proponuje przeprowadzenie szeregu eksperymentów z wykorzystaniem prostej przystawki i również prostych programów w języku Basic. Nie potrzebna będzie nawet umiejętność programowania w Basicu – do przeprowadzenia najprostszyc prób wystarczy mieć dostęp do jakiegokolwiek PC-ta i skorzystać ze wskazówek zawartych w artykule.

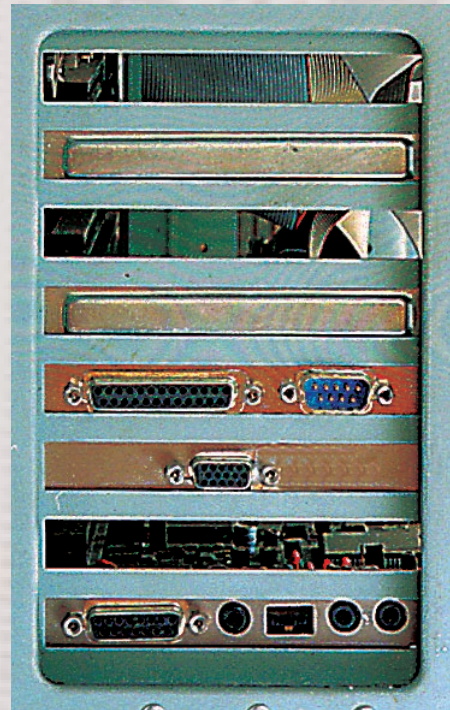
Każdy komputer musi współpracować z jakimiś urządzeniami zewnętrznymi. Bez takich urządzeń jak drukarka, mysz, joystick, modem telefoniczny, skaner, itp, sam komputer jest niemal zupełnie bezużyteczny.

Wymienione urządzenia dołączane są do komputera za pomocą tak zwanych portów. Wystarczy zajrzeć na tylną płytę obudowy komputera, by się przekonać, iż umieszczonych tam jest przynajmniej kilka gniazd połączeniowych. Umożliwiają one wysyłanie i przyjmowanie informacji.

Spośród gniazd znajdujących się na tylnej ścianie komputera, elektroników najbardziej interesują:

- porty szeregowy, oznaczane COM, itd, które są zgodne ze standardem RS-232, a które potocznie nazywamy komami, eresami lub serialami,
- port równoległy, zgodny ze standardem Centronics, oznaczany LPT, zwany potocznie elpetem,
- port joysticka, oznaczany GAME PORT, nazywany gejmporciem.

W każdym komputerze PC występują przynajmniej dwa porty szeregowy (COM1 i COM2). Komputer ma też przynajmniej jeden port równoległy (LPT1) i jeden port joysticka. Port joysticka ma specyficzną budowę i funkcje. W zasadzie służy tylko do przyjmowania informacji od współpracującego urządzenia (joysticka). Jego funkcje i możliwości wykorzystania zostaną omówione w dalszej części cyklu.



Zadaniem portów jest przesyłanie informacji na zewnątrz komputera oraz pobieranie informacji z zewnątrz, do i od urządzeń współpracujących.

Niemal wszystkie informacje przesyłane do i z komputera mają postać cyfrową, to znaczy, że mają postać napięcia odpowiadającego stanom logicznym 0 i 1.

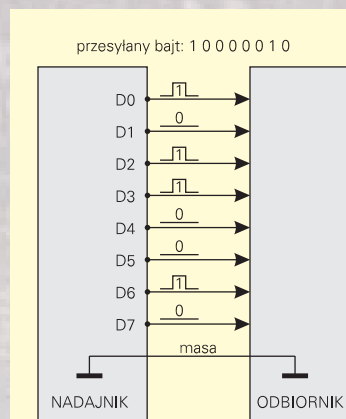
Zazwyczaj przesyłane są duże ilości informacji. Z różnych względów, od dawna podstawową „porcją” informacji jest bajt, czyli osiem bitów.

Informacje cyfrowe, jako kolejne bajty, mogą być przesyłane w postaci równoległej; wtedy potrzebne jest tyle linii – przewodów, ile bitów przesyłanych jest jednocześnie (plus przewód masy). Ilustruje to **rysunek 1a**.

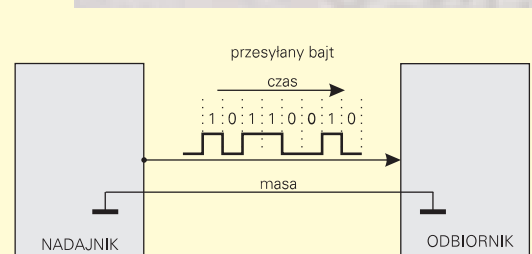
Informacje mogą też być przesyłane w postaci szeregowy, czyli kolejno jeden bit za drugim. Ilustruje to **rysunek 1b**. Tu wystarczy tylko dwa przewody – sygnałowy i masa.

Oba te sposoby wykorzystano do komunikacji komputera z otoczeniem.

Najprostsze jest działanie portu równoległego. Wykorzystano tu pierwszy sposób. W porcie tym między innymi znajduje się osiem linii, które pracują jako



Rys. 1a. Równoległe przekazywanie danych.



Rys. 1b. Szeregowy przesyłanie danych.

wyjścia. Komputer jednocześnie wysyła na te osiem wyjść cały bajt informacji. Ponieważ jednocześnie za pomocą ośmiu przewodów przesyłanych jest osiem bitów, transmisja danych przez port równoległy jest szybka. Oprócz ośmiu głównych linii wyjściowych, port równoległy zawiera szereg innych linii, które mogą służyć jako wejścia, a niektóre jako wyjścia. Są one potrzebne do wymiany informacji pomocniczych, na przykład sygnału synchronizującego, wskazującego, kiedy na liniach przesyłowych pojawia się nowy bajt informacji, czy informacji o błędach.

Port równoległy zazwyczaj obsługuje drukarkę, dlatego w praktyce do różnorodnych celów „elektronicznych” wykorzystuje się port szeregowy. Spośród dwóch portów szeregowych, jeden zwykle obsługuje myszkę, a drugi port może być wykorzystany w dowolny sposób. Dodatkową, cenną zaletą portu szeregowego jest fakt, że stanom logicznym 0 i 1 odpowiadają w nim napięcia +12V i -12V, i że z wyjść można pobrać prąd do 10mA. Tak znaczne napięcia i prądy są często wykorzystywane do... zasilania współpracujących urządzeń. W ten sposób wyjścia portu szeregowego stają się źródłem zasilania. Wyjścia i wejścia portu szeregowego są odporne na uszkodzenia pod wpływem zewnętrznych napięć i prądów, dlatego zasadniczo można dołączać do tego portu współpracujące urządzenia także przy włączonym zasilaniu komputera.

Wspomniane zalety zdecydowały, że port szeregowy jest najczęściej wykorzystywany przez konstruktorów do podłączania różnych bardziej lub mniej standardowych urządzeń zewnętrznych. Trzeba jednak mieć świadomość, że port szeregowy jest znacznie wolniejszy od portu równoległego, a więc jeśli wymagane jest szybkie przesyłanie danych, konieczne może się okazać wykorzystanie portu równoległego.

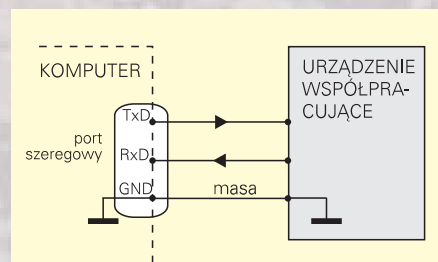
Właściwości portu szeregowego związane są ze starymi standardami stosowanymi w dalekopisach oraz z sygnałami wykorzystywanymi w modemach, czyli urządzeniach współpracujących z linią telefoniczną. Opis pełnego standardu jest bardzo obszerny. W opisie spotkasz tajemnicze określenia RS-232C, V.24, DTE, DCE, DSR, DTR, RTS, CTS, i wiele innych. Różnorodne możliwości połączeń, całe mnóstwo stosowanych kabli, tajemnicze skróty – wszystko to może wręcz przerazić przeciętnego użytkownika. Na szczęście do praktycznego wykorzystania portów komputera wystarczy garść najważniejszych informacji, natomiast szczegóły nie są potrzebne, dlatego zostaną pominięte.

W dalszej części artykułu przedstawione zostanie wszystko to, co jest potrzebne elektronikowi do praktycznego wykorzystania portu dla celów sterowania i zbierania danych. Ogólnie biorąc, poniższy materiał jest zgrubnym opisem standardu RS-232C.

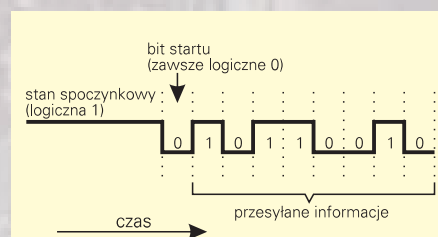
Port szeregowy

Główne dwie linie portu szeregowego oznaczone są TxD i RxD. TxD to linia Transmit Data, czyli wyjście. RxD (Receive Data) to wejście. Jak pokazuje **rysunek 2a**, te dwie linie plus przewód masy wystarczą, by przesyłać dane z i do komputera. Zastosowano tu tak zwaną transmisję szeregową, asynchroniczną. Urządzenie nadawcze wysyła kolejno w linię poszczególne bity. W największym uproszczeniu wygląda to jak na rysunku 1b. Przebieg napięcia o takiej postaci jest wysyłany w linię i odbierany przez współpracujące urządzenie. Ale taki ciąg bitów mógłby być błędnie odebrany, albo w skrajnym przypadku, zupełnie nie odebrany. Przykładowo przy transmisji samych zer lub samych jedynek urządzenie odbiorcze nie wiedziałoby, kiedy zaczyna się, a kiedy kończy transmisja. Dlatego przy szeregowym przesyłaniu danych trzeba wprowadzić dodatkowe informacje i spełnić pewne warunki. Przede wszystkim trzeba przyjąć jakiś stan spoczynkowy. Niech to będzie stan wysoki. Jeśli przez linię nie są przesyłane dane, na linii przesyłowej utrzymuje się stan wysoki. Pojawienie się stanu niskiego jest sygnałem o rozpoczęciu transmisji. Ale pierwszy bit transmitowanej informacji wcale nie musi być zerem. Dlatego konieczne jest wprowadzenie dodatkowego, początkowego bitu, który zawsze będzie zerem. Jedynym zadaniem tego bitu jest wskazanie odbiornikowi, że właśnie rozpoczyna się transmisja. Ten bit jest zwany bitem startu. Ilustruje to **rysunek 2b**.

Wydawałoby się, że po wysłaniu bitu startu, można potem przesłać kolejno dowolnie wielką ilość bitów. W praktyce wcale nie jest to takie proste. Odbiornik powinien wiedzieć, kiedy w linii pojawił się następny bit. A skąd ma to wiedzieć? Jeśli wykorzystana byłaby dodatkowa linia, przez którą przesyłany byłby sygnał taktujący, czyli zegarowy, nie byłoby żadnego problemu. Sygnał taktujący wskazywałby, kiedy w linii danych pojawia się następny bit (taki sposób nazywany jest transmisją synchroniczną). Ale tu nie ma sygnału zegarowego – zarówno nazwa, jak i rysunek 2 wskazują, iż jest to transmisja asynchroniczna, czyli nie ma żadnych dodatkowych linii przesyłających sygnały synchronizacji.



Rys. 2a. Najprostsze wykorzystanie portu szeregowego.



Rys. 2b. Przebiegi czasowe przy transmisji szeregowej.

Prędkość transmisji

Bit startu wskazuje tylko początek transmisji, a następne bity muszą pojawić się w ściśle określonym czasie. Inaczej mówiąc, zarówno nadajnik, jak i odbiornik muszą mieć równo tykające zegary, które odmierzą czas pojawiania się kolejnych bitów. Oznacza to, że nadajnik i odbiornik muszą się „umówić”, z jaką prędkością są przesyłane dane. Teoretycznie, prędkość przesyłania mogłaby być dowolna, jednak dla porządku przyjęto pewien standard. Dawniej stosowano bardzo małe prędkości transmisji: 50, 75 lub 110 bodów, czyli bitów na sekundę. Potem stosowano prędkości 300, 600, 1200, 2400, 3600 i 4800 bitów na sekundę. Obecnie w zależności od rodzaju współpracujących urządzeń, stosuje się prędkości transmisji 9600, 14400, 19200, 28800 bitów na sekundę i wyższe.

Zasadą jest, że w linię wysyła się kolejne bity danego bajtu, począwszy od bitu najmłodszego, oznaczanego D0 do najstarszego.

Przyjęcie standardowych prędkości transmisji znakomicie ułatwia współpracę urządzeń różnych producentów – wystarczy, by zachowane były ogólne zasady, w tym podane prędkości, czyli częstotliwości taktujące nadajnika i odbiornika.

A z jaką dokładnością należy utrzymywać podane prędkości transmisji i częstotliwości taktujące?

Jeśli zegary taktujące nadajnika i odbiornika tykałyby idealnie równo, po bicie startu można byłoby przesłać dowolną ilość bitów informacji bez obawy o błędy. Pokazuje to **rysunek 3**. Generator taktujący odbiornika wyznacza momenty czasu, które powinny wypadać dokładnie w połowie cza-

su przesyłania danego bitu. Tylko w tych momentach odbiornik próbkuje stan linii, czyli odczytuje aktualny stan linii. Momenty te na rysunku 3 zaznaczono strzałkami.

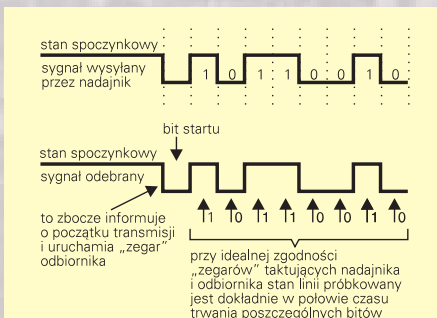
Jednak już przy jakiejś niewielkiej różnicy częstotliwości zegarów taktujących, dłuższy przekaz zostałby zinterpretowany błędnie. Jeśli częstotliwości zegarów będą się znacznie różnić, któryś kolejny moment próbkowania wypadnie w czasie trwania następnego lub poprzedniego bitu. Ilustruje to **rysunek 4**. Strzałki wskazują momenty próbkowania.

Wskutek różnicy częstotliwości zegarów taktujących nadajnika i odbiornika, przesyłana informacja została odczytana błędnie.

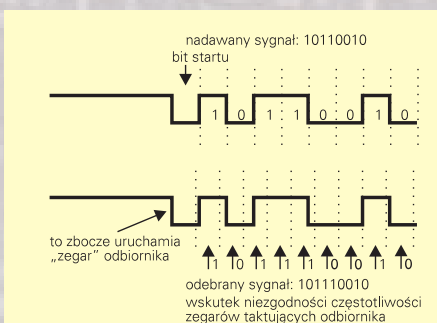
Wiadomo, że w praktyce idealnej dokładności zapewnić się nie da. Dla bezpieczeństwa przyjęto więc, że po bicie startu przesyła się tylko 5...8 bitów właściwej informacji. Wtedy wymagania na dokładność częstotliwości taktującej nie są zbyt ostre.

Ponieważ po bicie startu przesyłanych jest tylko kilka bitów informacji, urządzenia będą poprawnie pracować, nawet jeśli częstotliwości taktujące będą różnić się o kilka procent.

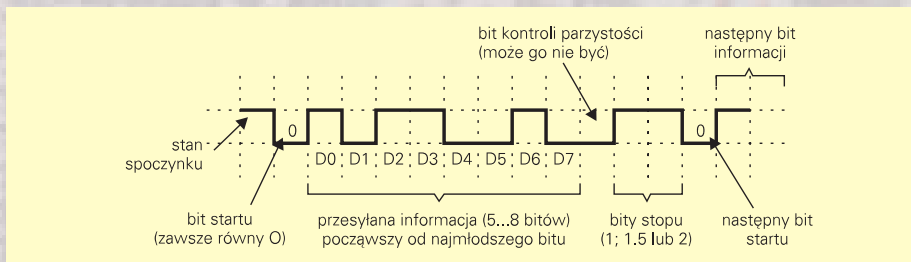
W praktyce, w urządzeniach standardu RS-232 stosuje się sygnał pomocniczy o częstotliwości 16 razy większej, niż standardowa częstotliwość przesyłania danych. Właśnie ten pomocniczy sygnał wyznacza momenty próbkowania. Ponieważ w czasie trwania każdego bitu wy-



Rys. 3. Sytuacja przy jednakowych częstotliwościach zegarów nadajnika i odbiornika.



Rys. 4. Sytuacja przy niejednakowych częstotliwościach zegarów nadajnika i odbiornika



Rys. 5. Struktura informacji przy przesyłaniu danych łączem szeregowym RS-232.

stępuje 16 taktów tego sygnału pomocniczego, próbkowanie „środku bitu” odbywa się w ósmym takcie każdego cyklu.

Częstotliwości taktujące, potrzebne przy różnych prędkościach transmisji, uzyskuje się z jednego generatora kwarcowego przez odpowiedni podział jego częstotliwości.

Omówiliśmy już dwie sprawy związane z transmisją szeregową: znaczenie bitu startu oraz dokładność częstotliwości taktujących nadajnika i odbiornika.

Trzecią sprawą jest bit kontroli parzystości.

Bit kontroli parzystości

Przy przesyłaniu danych na odległość, chwilowe zakłócenia indukujące się w przewodach, albo też inne szkodliwe czynniki, mogą wprowadzić błędy i odebrany sygnał będzie różnić się od nadanego. Prawdopodobieństwo wystąpienia błędu jest w sumie niewielkie, ale nie można go wykluczyć. Dobrze byłoby mieć informację o wystąpieniu błędów w transmisji. Wtedy dane można przesłać ponownie.

Wprowadzono więc dodatkowy bit kontrolny. Wartość tego bitu zależy od przesyłanej informacji. Umawiamy się, że transmitowane dane powinny zawierać parzystą liczbę jedynek. Jeśli akurat przesyłane kilka bitów zawiera parzystą liczbę jedynek, bit kontrolny dodawany w nadajniku ma wartość zero. Jeśli właściwa informacja zawiera nieparzystą liczbę jedynek, bit kontrolny ustawiany jest na 1, aby całkowita liczba jedynek była parzysta. Po stronie odbiorczej sprawdzana jest ilość odebranych jedynek. Jeśli ich liczba nie jest parzysta, odbiornik sygnalizuje błąd. Przy odpowiedniej organizacji transmisji, po wykryciu błędu zafalszowane dane zostaną przesłane jeszcze raz.

Co prawda wprowadzenie bitu kontroli parzystości nie daje gwarancji wykrycia wszystkich błędów. Na przykład przy jednoczesnym zaistnieniu dwóch przekłamań odbiornik nie wykryje błędu – jednak prawdopodobieństwo takiego zdarzenia jest o wiele mniejsze, niż szansa pojawienia się jednego przekłamań i w praktyce to wystarcza.

Notujemy kolejną ważną wiadomość: w przesyłanym sygnale może się zna-

leźć, choć nie musi, dodatkowy bit kontroli parzystości.

Bit Stopu

Po wysłaniu jednej porcji informacji nadajnik ustawi na linii stan spoczynkowy, czyli stan wysoki. Jeśli nadajnik miał wysłać tylko te kilka bitów (jeden bajt), to stan wysoki będzie się utrzymywał, aż po upływie dowolnie długiego czasu nadajnik otrzyma rozkaz wysłania następnej porcji informacji (następnego bajtu). W praktyce często trzeba przesłać dużą ilość danych i wtedy nadajnik wysyła porcje informacji jedną po drugiej. Między poszczególnymi porcjami musi wystąpić przerwa, aby odbiornik mógł prawidłowo odebrać bit startu następnej porcji. Ta przerwa musi trwać przynajmniej przez czas odpowiadający transmisji jednego bitu (albo 1,5, albo 2 bitów). To właśnie jest kolejny warunek.

Już wiemy, że przy transmisji szeregowej łączem RS-232, oprócz 4...8 bitów właściwej informacji, przesyłany jest bit startu, bit(y) stopu i ewentualnie bit kontroli parzystości.

Wszystkie te zasady, czyli tak zwany protokół transmisji, mogą się wydać skomplikowane. Ale obecnie nikt nie próbuje zbudować układu realizującego podane funkcje z kostek TTL czy CMOS4000. W praktyce albo o wszystko troszczy się mikroprocesor, albo wykorzystywane są specjalizowane układy scalone, tak zwane UARTy (Uniwersal Asynchronous Receiver Transmitter). W PC-tach są to kostki 8250, 16450 lub ich odpowiedniki. Istnieje też wiele innych UARTów, na przykład 8251, IM6402 (IM6403), itp.

Użytkownik nie troszczy się o szczegóły. Musi tylko poinformować taką kostkę jaka będzie prędkość transmisji, ile bitów będzie transmitowanych w jednej porcji (5...8), czy wystąpi bit kontroli parzystości (E – even), nieparzystości (O – odd), czy nie będzie takiego bitu (N – no parity) oraz jaki jest minimalny czas przerwy między kolejnymi porcjami (1, 1,5 lub 2 bity stopu). Przy obsłudze komputera zwykle ustala się te parametry programowo.

Ostatecznie struktura przesyłanych informacji jest taka, jak na **rysunku 5**.

(red)

Cd. w EdW7/97