

Interfejsy w Systemach Komputerowych - ULTIMATE

SonMati

Ervelan

Doxus

23 grudnia 2014

Pytania i odpowiedzi

1 RS-232

Prawda/Falsz

- RS-232 jest portem przeznaczonym do synchronicznej transmisji znakowej. Generator taktu odpowiedzialny za wyprowadzanie znaków typowo ustawiany jest na: 1200bd, 2400bd, 4800bd, 9600bd, 19200bd.
RS-232 jest portem przeznaczonym do asynchronicznej transmisji znakowej. Da się sztucznie stworzyć synchroniczną transmisję.
- Linie kontrolne w interfejsie RS-232 to: DTR, DSR, RTS, CTS, RI, DCD. Pary DTR/DSR i RTS/CTS wykorzystywane są do realizacji handshake'u w połączeniach bezmodemowych.
Tak, te pary linii mogą być wykorzystywane do handshake podczas gdy RxD i TxD zajmują się przesyłem danych.
- Transakcja w systemie MODBUS składa się z zapytania (query) wysyłanego przez stację Slave i odpowiedzi odsyłanej przez stację Master.
Jest odwrotnie - zapytanie wysyła Master, a odpowiedź odsyła Slave.
- W trybie transmisji ASCII znacznikiem początku ramki jest znak ':', a kooca ramki para znaków CR LF. W trybie transmisji RTU znacznikiem początku ramki jest znak 'Ctrl-A', a kooca para znaków CTRL-Y CTRL-Z.
Zdanie jest poprawne dla ASCII. Dla RTU, znacznikiem początku i końca ramki jest przerwa o długości minimum $4T$, gdzie T jest czasem trwania jednego znaku.
- Standard RS-232 transmituje znaki synchronicznie, bity w znakach [asynchronicznie]
Ostatnie słowo ucięte, więc spekuluję że tak właśnie było napisane. To nieprawda, jest odwrotnie.
- Standard RS-422 pozwala na osiągnięcie szybkości 10MBodów na odległości 100m.
IMO pozwala, na slajdzie 12 jest napisane że 10 Mbd przy zasięgu DO 100m - czyli 100m chyba też.
- Liniami kontrolnymi w RS-232 nie są linie TxD, RxD, SG.
Owszem, TxD i RxD są liniami danych, a SG to po prostu masa.
- System MODBUS składa się z faz zapytania i odpowiedzi.
Tak właśnie jest.
- W systemie MODBUS
 - Obowiązuje master/slave.
Pewnie, a w dodatku Slave'ów może być wielu.
 - Prędkości transmisji wynoszą od 1200 do 19200bd.
Jak najbardziej.
 - Ramka w ASCII może mieć format 7N2 (lub np. 7E1, 7O1).
Tak, patrz warstwa fizyczna MODBUS.
 - Ramka w RTU może mieć format 8N2 *(lub np. 8E1, 8O1).
Tak, patrz warstwa fizyczna MODBUS.
- W trybie transmisji RTU jest kontrola błędów CRC.
Tak, jest elementem budowy ramki RTU.
- Bit kontrolny w RS-232 zależy od bitu danych i bitu stopu.
Bit kontrolny służy do kontroli parzystości/nieparzystości, nie ma związku z bitem stopu.
- Za pomocą RS-232 możemy połączyć ze sobą 2 stacje DCE
Połączyć możemy dwie stacje DTE, lub DTE z DCE. Dwie stacje DCE łączą się za pomocą łącza telefonicznego.

- W MODBUS kontrola błędów jest realizowana za pomocą LRC lub CRC.
Tak, LRC wykorzystywane jest w trybie ASCII, CRC w trybie RTU.
- Do portu RS 485 można podłączyć tylko jedno urządzenie, ale za to obsługiwać go z dużo większą szybkością i na większą odległość niż jest to możliwe w przypadku interfejsu RS 232.
Można podłączyć do 32 stacji.
- Format ramki w protokole Modbus jest następujący: znacznik początku ramki, adres urządzenia slave, adres mastera, pole danych, znacznik końca ramki.
Opis nie pasuje ani do trybu ASCII, ani RTU
- RS 232 jest portem przeznaczonym dla asynchronicznej transmisji znakowej, realizowanej zazwyczaj w trybie dwukierunkowym, czyli dwukierunkowej transmisji niejednoczesnej (naprzemiennej)
Tryb dwukierunkowy jest równoczesny, to półdwukierunkowy jest niejednoczesny.
- W interfejsie RS 232 linie TxD i RxD służą do transmisji znaków, natomiast DTR, RTS to wyjścia kontrolne, a DSR, CTS, RI i DCD to wejścia kontrolne.
Indeed
- Multipleksowanie urządzeń ze znakowym portem asynchronicznym pozwala na ich kontrolę poprzez jeden port RS-232.
Żeby kontrolować kilka urządzeń z jednego portu potrzebny jest koncentrator. Jeśli "używanie koncentratora" równa się "multipleksowanie", to PRAWDA.
- Węzeł podrzędny w systemie MODBUS po wykryciu błędu w komunikacji wysyła potwierdzenie negatywne do węzła nadrzędnego.
W odpowiedzi pole to jest wykorzystywane do pozytywnego lub negatywnego potwierdzenia wykonania polecenia.
- Czy w trybie ASCII systemu MODBUS każdy bajt wysyłany jest jako znak z przedziału 0x00, 0xFF?
Bajt dzielimy na 2 części i wysyłamy jako 2 znaki z przedziału 0-9 i A-F

2 USB

Prawda/Fałsz

- Kontrola urządzenia USB odbywa się poprzez zapisy komunikatów do bufora o numerze 0 i odczyty informacji statusowych z bufora o numerze 0.
Zgadza się.
- W przypadku błędu transmisji każda transakcja USB jest powtarzana, ponieważ niedopuszczalne jest przekazywanie danych przekłamanych.
Transakcje izochroniczne nie są powtarzane w przypadku błędu transmisji.
- Hub nie dopuszcza ruchu full speed do portów, do których są podłączone urządzenia low speed.
Tak, urządzenie lowspeed blokuje możliwość włączenia fullspeed na całym porcie.
- Reset portu USB polega na rekonfiguracji hosta, po której host zapisuje tablicę deskryptorów do urządzenia podłączonego do tego portu.
Reset portu USB polega na rekonfiguracji urządzenia. W następującej procedurze enumeracji między innymi dochodzi do odczytu tablicy deskryptorów z urządzenia przez host.
- Typowa transakcja USB składa się z pakietów żądania i odpowiedzi, z których każdy potwierdzany jest osobnym potwierdzeniem.
Typowa transakcja USB składa się z pakietów token, data i handshake. Transakcje izochroniczne nie są potwierdzane.

- W systemie USB urządzenia zgłaszają żądania do hosta, który je kolejkuje i następnie obsługuje w kolejności pojawiania się zgłoszenia.
Urządzenia nie zgłaszają żądań, tylko są odpytywane przez hosta. Host nie tworzy jednej kolejki, tylko w miarę możliwości stara się obsługiwać wszystkie urządzenia jednocześnie, równomiernie, zapobiegając zawłaszczeniu.
- W USB można połączyć kaskadowo do 5 hubów, korzystających z zasilania magistralowego
Podłączyć je można tylko korzystając z zasilania zewnętrznego lub hybrydowego. Przy zasilaniu magistralowym zabraknie zasilania już na drugim hubie. Co więcej, należy mieć na uwadze maksymalne dopuszczalne opóźnienie sygnału, które przy przejściu przez 5 hubów jest osiągane - 350ns. Urządzenia podpięte do 5'tego huba mogą nie działać poprawnie.
- Mechanizm data toggle w USB służy do przywracania synchronizacji pomiędzy hostem i urządzeniem, utraconej na skutek wystąpienia błędów w pakietach danych.
Mechanizm data toggle zabezpiecza przed utratą synchronizacji pomiędzy hostem i urządzeniem na skutek błędów w potwierdzeniu odsyłanym przez odbiorcę.
- Host kontroler USB komunikuje się z interfejsem magistrali USB urządzenia peryferyjnego za pomocą fizycznego kanału komunikacyjnego.
Tak, używamy kabelka.
- Kamera internetowa może przysyłać obraz do komputera za pomocą transferu izochronicznego z szybkością LowSpeed w interfejsie USB.
Z tabelki można wyczytać, że dla transferu izochronicznego nie można wykorzystać szybkości LowSpeed.
- Pakiety USB przesyłane z szybkością LowSpeed muszą być poprzedzone pakietem preambuły
Tak, jest on charakterystyczny dla pakietów przesyłanych z szybkością LowSpeed
- Urządzenie peryferyjne USB 2.0 może być podłączone do host kontrolera za pośrednictwem maksymalnie sześciu hubów.
Aby spełnić normę (ograniczenie czasowe oczekiwania na odpowiedź), można podłączyć za pośrednictwem maksymalnie 5 hubów.
- Pole PID w pakiecie USB zabezpieczone jest 16-bitową sumą kontrolną CRC.
Pole PID zabezpieczone jest 4-bitowym polem kontroli, będącym prostą negacją bitów pola PID.
- Do portu dolnego huba podłączane mogą być tylko wtyki USB typu B.
Tylko wtyki typu A.
- Transakcja dzielona w USB 1.1 składa się z dwóch części: SSPLIT i CSPLIT.
Takie czary dopiero w USB 2.0
- W przypadku połączenia USB HighSpeed wykonywane jest podparcie linii D- do Vcc za pośrednictwem rezystora 1,5k.
Po podłączeniu urządzenia High Speed w pierwszej kolejności jest ono identyfikowane jako Full Speed, więc wykonywane jest podparcie linii D+ do Vcc za pośrednictwem rezystora 1,5k. Następnie, poprzez chirp ("dzwierkanie") host i urządzenie ustalają, czy możliwa jest komunikacja w trybie High Speed. Jeśli tak, usuwane jest podparcie przez rezystor, a obwód zamykany jest terminatorami.
- W kodowaniu NRZI co sześć jedynek jest wstawiany bit synchronizacji "0".
Pomieszczone pojedynczo. W kodowaniu NRZI nie występuje dodawanie bitu synchronizacji. Proces ten nazywa się bit stuffing. Zdanie byłoby poprawne, gdyby brzmiało np. W kodowaniu NRZI z bit stuffingiem co sześć.
- Transakcje kontrolna i przerwaniowa w USB 1.1 są transakcjami aperiodycznymi z gwarantowanym pasmem w ramach jednej mikroramki.
Transakcja kontrolna jest transakcją aperiodyczną. Transakcja przerwaniowa jest transakcją periodyczną.
- W kontrolerze OHC transakcje izochroniczne są porządkowane/kolejkowane w drzewo/strukturę drzewiastą.
Tak, OHC wykorzystuje strukturę drzewa, a UHC tablicę wskaźników (listę podwieszoną).

- **Standard USB 2.0 wymaga skręconych, ekranowanych kabli.**
Well, High speed all the way, więc wymaga
- **Transfer kontrolny i przerwaniowy są transferami aperiodycznymi.**
Było podobne pytanie. Transfer kontrolny jest aperiodyczny, transfer przerwaniowy jest periodyczny.
- **Wielowarstwowa architektura USB 2.0 składa się z 3 warstw.**
Tak - warstwa interfejsu magistrali USB, warstwa urządzenia USB, warstwa funkcji urządzenia
- **W porcie USB dane są dzielone na transakcje.**
Dane w ramce są dzielone na transakcje, więc tak
- **Hub podłączony do portu USB ma obciążalność 100mA.**
Hub podłączony do portu USB bez własnego zasilania (zasilanie magistralowe) ma obciążalność dla portów dolnych do 100mA na port (maksymalną 400mA na cały hub). Hub z zasilaniem zewnętrznym lub hybrydowym ma obciążalność do 500mA na port.
- W systemie USB do mechanizmów kontroli danych należą:
 - **Przełączanie pakietów danych**
Tzw. Data Toggle
 - **Wykrywanie braku aktywności na linii danych;**
 - **Zabezpieczenie znacznika SOF lub EOF**
Reakcją jest natomiast objęcie wystąpienia fałszywego znacznika końca pakietu (false EOP)
 - **kodowanie LRC**
Pakiety zabezpieczone są kodowaniem CRC.
- **Wydajność dolnego portu (USB 2.0) wynosi 500mA.**
Nie wiadomo. Zasilany Hub może wystawić te 500mA, ale niezasilany już tylko 100mA
- **USB 2.0 ma parę przewodów ekranowanych.**
Taki upgrade.
- **W kodowaniu NRZ wstawia się dodatkowe bity synchroniczne.**
Dodatkowe bity synchroniczne wstawia się w kodowaniu NRZI
- **Urządzenie USB 2.0 może zasygnalizować swoją niegotowość do zapisu danych z szybkością High-Speed wysyłając pakiet PING-NYET.**
Wychodzi na to, że niegotowość zgłasza samym NYET? Pyta – PING, odpowiada (niegotowość) NYET. I Tak cały czas, chyba że dostanie ACK. ACK – wykonanie transakcji OUT. NYRT – host kontynuuje wysyłanie zapytań PING
- **W systemie deskryptorów urządzenia USB może wystąpić kilka deskryptorów urządzenia, konfiguracji, interfejsów i punktów końcowych.**
Deskryptor urządzenia może być jeden. Innych – konfiguracji, interfejsu, końcowych może być więcej.
- **Hub USB ma przerwaniowy punkt końcowy, który wykorzystuje do powiadamiania hosta o podłączeniu urządzenia USB do którego z jego portów dolnych.**
Chyba.
- **Na wierzchołku wielopoziomowego, hierarchicznego układu deskryptorów USB znajduje się deskryptor konfiguracji. Na szczycie znajduje się pojedynczy deskryptor urządzenia.**
- **Transfer masowy i izochroniczny USB 1.1 są przykładami transferów aperiodycznych z zagwarantowanym pasmem w ramach jednej mikroramki.**
Izochroniczny jest periodyczny, masowy nie ma zagwarantowanego pasma (wg tabelki z prędkościami)
- **W deskryptorze konfiguracji USB jest jakiś pole statusowe, które mówi o maksymalnym poborze prądu. Dla wartości 50 urządzenie pobiera 50mA.**
Pole to jest tak skonstruowane, żeby wartość zmieściła się w jednym bajcie, ze skokiem co 2mA. Dlatego urządzenie, które zgłasza, że 50 może zasysać maksymalnie 100mA.
- **Uszeregowanie transakcji w USB. Nie zależy od implementacji kontrolera. w OHC przerwaniowe są w strukturze drzewa, a w UCH listy podwieszanej, co ma wpływ na uszeregowanie (do sprawdzenia)**

3 IEEE 1394 Firewire

4 IEEE-488 i SCPI