helion.pl
KSIĘGARNIA INTERNETOWA · WYDAWNICTWO

IDŹ DO
PRZYKŁADOWY ROZDZIAŁ
SPIS TREŚCI

KATALOG KSIĄŻEK
KATALOG ONLINE
ZAMÓW DRUKOWANY KATALOG

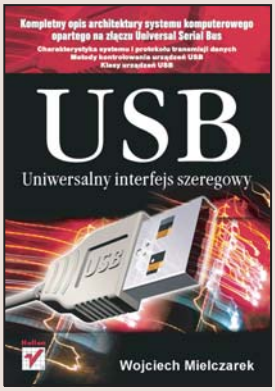
TWÓJ KOSZYK
DODAJ DO KOSZYKA

CENNIK I INFORMACJE
ZAMÓW INFORMACJE O NOWOŚCIACH
ZAMÓW CENNIK

CZYTELNIA
FRAGMENTY KSIĄŻEK ONLINE

USB. Uniwersalny interfejs szeregowy

Autor: Wojciech Mielczarek
ISBN: 83-246-0142-2
Format: B5, stron: 128



Kompletny opis architektury systemu komputerowego opartego na złączu Universal Serial Bus

- Charakterystyka systemu i protokołu transmisji danych
- Metody kontrolowania urządzeń USB
- Klasy urządzeń USB

Złącze USB (Universal Serial Bus) szybko stało się jednym z podstawowych standardów komunikacyjnych w systemach komputerowych, zastępując wiele dotychczas stosowanych portów. Dzięki kilku typom transferu danych i szybkości transmisji umożliwia przyłączenie różnych urządzeń peryferyjnych. Poprzez USB podłącza się między innymi drukarki, skanery, aparaty cyfrowe, klawiatury oraz myszy. Wykrywanie i konfiguracja urządzenia USB oraz instalacja sterownika odbywa się automatycznie, co bardzo ułatwia korzystanie z osprzętu zgodnego z tym standardem. Niestety, łatwość obsługi urządzeń USB nie idzie w parze z prostotą samego interfejsu. Mechanizmy komunikacyjne USB są skomplikowane, a wykonywanie układów interfejsów dla urządzeń USB oraz pisanie sterowników urządzeń dla kontrolera systemu to trudne zadania.

Książka „USB. Uniwersalny interfejs szeregowy" to omówienie architektury Universal Serial Bus. Przeznaczona jest dla osób, które chcą zmierzyć się z takim wyzwaniem lub po prostu są zainteresowane tym, co „w komputerze piszczy". Opisany jest tu standard USB 1.1 oraz jego rozszerzenia wprowadzone w wersji 2.0, omówione są mechanizmy komunikacyjne zastosowane w USB oraz specyfika klas urządzeń USB. Poznasz też protokół transmisji danych przez złącze USB, nauczysz się kontrolować urządzenia za pomocą rozkazów standardowych oraz dowiesz się, jak tworzyć oprogramowanie obsługujące port USB.

- Model komunikacyjny interfejsu USB
- Transmisja danych i wykrywanie błędów
- Deskryptory urządzeń USB
- Wykrywanie podłączonych urządzeń
- Standardowe rozkazy protokołu USB
- Zasilanie urządzeń
- Klasy urządzeń USB

Jeśli chcesz wiedzieć, jak działa interfejs USB i jak można go wykorzystać — koniecznie przeczytaj tę książkę.

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

Spis treści

Wstęp 7

Rozdział 1. Charakterystyka systemu USB 9

1.1. Podstawowe właściwości interfejsu USB 10
„Gorące" podłączenie 10
Jeden typ złącza 10
Duża liczba podłączanych urządzeń 10
Różne szybkości transmisji 10
Zasilanie 11
Protokół komunikacyjny, detekcja błędów 11
Transfery USB 11
Zasoby systemowe wymagane przez USB 11
Koszt 11
1.2. Środowisko sygnałowe i fizyczne interfejsu USB 12
Wykrywanie podłączenia lub odłączenia urządzenia 14
Kodowanie bitów w systemie USB 15
Środowisko fizyczne 16
1.3. Ramki i mikroramki 17
1.4. Model komunikacyjny 18
Elementy programowe USB po stronie hosta 19
Elementy sprzętowe USB po stronie hosta 20
Elementy programowe USB po stronie urządzenia 21
Elementy sprzętowe po stronie urządzenia 21
1.5. Transfery USB 22
1.6. Zarządzanie magistralą USB 24
1.7. Stany urządzenia USB 26
1.8. Hub w systemie USB 28

Rozdział 2. Protokół komunikacyjny 31

2.1. Pakiety w systemie USB 31
Pakiet preambuły 34
2.2. Transakcje USB 35
Transakcje przerwaniowe (interrupt transactions) 36
Transakcje masowe (bulk transactions) 36

Transakcje izochroniczne (isochronous transactions) ... 37
Transakcje kontrolne (control transactions) ... 39
2.3. Wykrywanie błędów i kontrola transmisji ... 41
Kontrola poprawności pakietów ... 41
Ograniczenie czasowe oczekiwania na odpowiedź ... 42
Przełączanie pakietów danych ... 43
Praca normalna — synchronizacja nadawcy i odbiorcy ... 44
Praca zakłócona — utrata synchronizacji nadawcy i odbiorcy i odtworzenie synchronizacji ... 44

Rozdział 3. Deskryptory w urządzeniach USB ... 47
3.1. Deskryptory w urządzeniach zgodnych z USB 1.x ... 48
3.2. Deskryptory w urządzeniach zgodnych z USB 2.0 ... 52

Rozdział 4. Wykrywanie i enumeracja urządzenia w systemie USB ... 55
4.1. Procedura enumeracji ... 55
4.2. Przełączanie segmentu USB na wysoką szybkość ... 57

Rozdział 5. Kontrola urządzenia USB za pośrednictwem rozkazów standardowych ... 59
5.1. Etapy transferu kontrolnego ... 60
Przekazanie rozkazu (Setup Stage) ... 60
Przekazanie danych (Data Stage) ... 61
Przekazanie statusu (Status Stage) ... 62
5.2. Standardowe rozkazy USB ... 62

Rozdział 6. Zasilanie urządzeń w systemie USB ... 71
6.1. Dystrybucja zasilania w USB ... 71
Hub zasilany z magistrali USB podłączony do portu o obciążalności 500 mA ... 71
Hub z zasilaniem hybrydowym ... 72
Hub całkowicie zasilany z zasilacza własnego ... 72
6.2. Zarządzanie zasilaniem w systemie USB ... 73
Urządzenie w stanie zawieszenia ... 74
Wznowienie normalnej pracy urządzenia ... 74

Rozdział 7. Konfiguracja huba ... 77
7.1. Deskryptor klasy hub ... 77
7.2. Charakterystyczne punkty końcowe i obsługa huba ... 80
Zanik zasilania lub przeciążenie prądowe huba ... 81
Zmiana na porcie dolnym ... 83

Rozdział 8. Klasy urządzeń USB ... 87
8.1. Klasa audio ... 89
8.2. Klasa HID ... 90
Rozkazy specyficzne dla klasy HID ... 93
Interfejs inicjalizacyjny w urządzeniach klasy HID ... 96

Rozdział 9. Oprogramowanie USB w hoście ... 97
9.1. Komponenty programowe USB ... 99
Sterownik urządzenia ... 99
Sterownik USB ... 100
Sterownik host kontrolera ... 100

9.2. Zarządzanie konfiguracją ... 101
Nadzór zasilania ... 101
Monitorowanie i przydział pasma ... 102
9.3. Zarządzanie magistralą ... 103
9.4. Zarządzanie transferem danych ... 103

Rozdział 10. Szczególne przypadki komunikacji w systemie USB 2.0 ... 105
10.1. Protokół PING-NYET ... 105
10.2. Transakcja SPLIT ... 106
10.3. Komunikacja z szerokopasmowym punktem izochronicznym ... 107
10.4. Kodowanie pakietów wprowadzonych w USB 2.0 ... 108

Rozdział 11. Interfejs USB w urządzeniach ... 111

Rozdział 12. Suplement On-The-Go do specyfikacji USB 2.0 ... 115
12.1. Właściwości mechaniczne i elektryczne łącza OTG ... 116
12.2. Protokół SRP ... 118
Warunki początkowe ... 119
Pulsowanie na linii danych ... 119
Czas trwania SRP ... 120
Odpowiedź urządzenia A na SRP ... 120
12.3. Protokół HNP ... 120
Deskryptor OTG ... 121

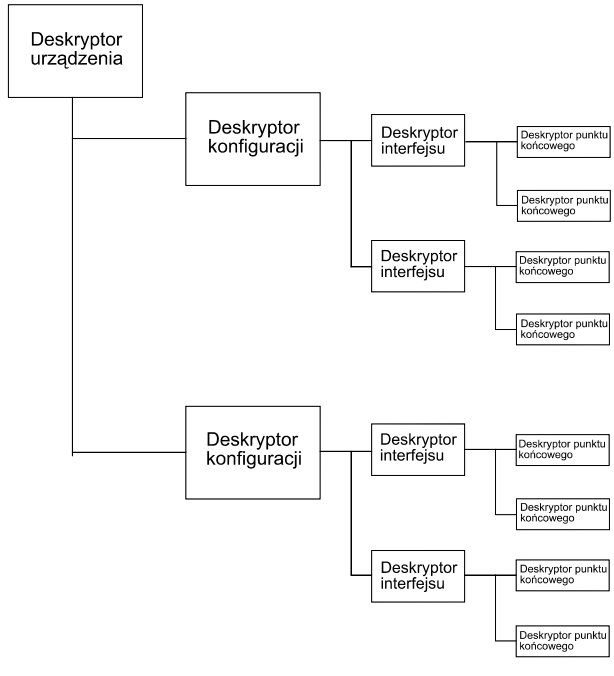
Literatura ... 123

Skorowidz ... 125

Rozdział 3.

Deskryptory w urządzeniach USB

W systemie USB możliwa jest komunikacja z urządzeniami o różnych właściwościach komunikacyjnych, przy czym właściwości komunikacyjne poszczególnych urządzeń muszą być znane hostowi zarządzającemu pracą systemu. Dlatego też w każdym urządzeniu USB znajduje się pełna informacja o sposobie komunikacji z urządzeniem udostępniana podczas procesu enumeracji. Informacja ta przechowywana jest w deskryptorach, które są tablicami o ściśle określonej strukturze. Deskryptory urządzenia USB tworzą hierarchiczną strukturę, na której wierzchołku znajduje się deskryptor urządzenia, jak to przedstawiono na rysunku 3.1.



Rysunek 3.1. *Hierarchiczna struktura deskryptorów w urządzeniu USB*

3.1. Deskryptory w urządzeniach zgodnych z USB 1.x

W każdym urządzeniu USB występuje tzw. deskryptor urządzenia (*device descriptor*), który między innymi zawiera informację o liczbie konfiguracji dostępnych w urządzeniu. W tabeli 3.1 przedstawiono znaczenie poszczególnych pól deskryptora urządzenia. Każdą konfigurację z kolei opisuje oddzielny deskryptor konfiguracji (*configuration descriptor*), który między innymi informuje o liczbie interfejsów przypisanych danej konfiguracji. W tabeli 3.2 przedstawiono znaczenie poszczególnych pól deskryptora konfiguracji. Każdy interfejs posiada własny deskryptor interfejsu (*interface desciptor*), który między innymi określa liczbę punktów końcowych związanych z tym interfejsem. W tabeli 3.3 przedstawiono znaczenie poszczególnych pól deskryptora interfejsu. Podobnie, każdy punkt końcowy charakteryzowany jest przez oddzielny deskryptor punktu końcowego (*endpoint descriptor*).

Tabela 3.1. *Deskryptor urządzenia*

Numer pola	Rozmiar pola (liczba bajtów)	Nazwa pola	Znaczenie i wartość
1	1	Długość deskryptora (*Descriptor Length*)	Liczba bajtów w deskryptorze urządzenia
2	1	Typ deskryptora (*Descriptor Type*)	Stała określająca typ deskryptora. Wartość 1 oznacza deskryptor urządzenia
3	2	Wersja USB (*USB version #*)	Liczba w kodzie BCD określająca numer wersji USB (np. 1.00)
4	1	Klasa (*Device Class*)	Kod klasy, do której należy urządzenie USB. Wartość pola równa 0 oznacza, że interfejsy w ramach danej konfiguracji mają własne kody klasy i działają niezależnie. Wartość 256 oznacza, że mamy do czynienia z urządzeniem specyficznym, którego klasa jest określona przez producenta.
5	1	Podklasa (*Device Subclass*)	Kod podklasy związany z kodem klasy urządzenia. Jeżeli kod klasy = 0, to i kod podklasy = 0
6	1	Protokół (*Device Protocol*)	Kod protokołu urządzenia. Protokół może być zawiązany: z daną klasą (podklasą) urządzenia i wtedy jest zdefiniowany przez standard danej klasy lub określony przez producenta (wartość pola: 256). Wartość pola równa 0 oznacza, że nie stosuje się protokołu określonego przez klasę urządzenia, jednak poszczególne interfejsy urządzenia mogą wykorzystywać protokoły klas, do których należą

Tabela 3.1. *Deskryptor urządzenia — ciąg dalszy*

Numer pola	Rozmiar pola (liczba bajtów)	Nazwa pola	Znaczenie i wartość
7	1	Maksymalny rozmiar pakietu (*MaxPacketSize*)	Maksymalny rozmiar pakietu danych przy komunikacji z punktem końcowym 0 (EP0). Dopuszczalne wartości: 8, 16, 32, 64 bajty
8	2	Identyfikator producenta (*Vendor ID*)	Identyfikator producenta przypisany mu przez standard USB
9	2	Identyfikator produktu (*Product ID*)	Identyfikator produktu określony przez wytwórcę
10	2	Numer wersji urządzenia (*Version number*)	Numer wersji urządzenia w kodzie BCD
11	1	Wytwórca (*Manufacturer*)	Indeks łańcucha znakowego określającego wytwórcę
12	1	Nazwa produktu (*Product Name*)	Indeks do łańcucha znakowego określającego urządzenie
13	1	Numer seryjny (*Serial Number*)	Indeks do łańcucha znakowego określającego numer seryjny urządzenia
14	1	Liczba konfiguracji (*Number of Configurations*)	Liczba możliwych konfiguracji urządzenia

Tabela 3.2. *Deskryptor konfiguracji*

Numer pola	Rozmiar pola (liczba bajtów)	Nazwa pola	Znaczenie i wartość
1	1	Długość deskryptora (*DescriptorLength*)	Liczba bajtów w deskryptorze konfiguracji
2	1	Typ deskryptora (*Descriptor Type*)	Stała określająca typ deskryptora. Wartość 2 oznacza deskryptor konfiguracji
3	2	Długość całkowita (*TotalLength*)	Liczba bajtów określająca daną konfigurację. Jest to łączna długość deskryptorów: konfiguracji, interfejsów, punktów końcowych oraz specyficznych dla klasy lub producenta, zwracana przez urządzenie dla danej konfiguracji
4	1	Liczba interfejsów (*NumInterfaces*)	Liczba interfejsów dla danej konfiguracji urządzenia
5	1	Numer konfiguracji (*Configuration Value*)	Liczba przypisana danej konfiguracji, stosowana jako argument rozkazu Set Configuration w celu wyboru tej konfiguracji
6	1	Indeks opisu konfiguracji (*Index*)	Indeks deskryptora łańcuchowego opisującego daną konfigurację

Tabela 3.2. *Deskryptor konfiguracji — ciąg dalszy*

Numer pola	Rozmiar pola (liczba bajtów)	Nazwa pola	Znaczenie i wartość
7	1	Atrybuty (*Attributes*)	Mapa bitowa charakteryzująca konfigurację: D7 — urządzenie zasilane z magistrali USB, D6 — urządzenie z własnym zasilaniem, D5 — dostępna funkcja Remote Wakeup, D4-0 — zarezerwowane, ustawione na 0
8	1	Maksymalna moc (*MaxPower*)	Maksymalny pobór prądu zasilania (w mA) z magistrali USB przez urządzenie pracujące w tej konfiguracji. Krokiem są tu 2 mA, stąd np. wartość 75 oznacza prąd 150 mA

Tabela 3.3. *Deskryptor interfejsu*

Numer pola	Rozmiar pola (liczba bajtów)	Nazwa pola	Znaczenie i wartość
1	1	Długość deskryptora (*DescriptorLength*)	Liczba bajtów w deskryptorze interfejsu
2	1	Typ deskryptora (*Descriptor Type*)	Stała określająca typ deskryptora. Wartość 4 oznacza deskryptor interfejsu
3	1	Numer interfejsu (*InterfaceNumber*)	Liczba identyfikująca indeks w tablicy interfejsów w danej konfiguracji
4	1	Ustawienie alternatywne (*AlternateSetting*)	Wartość wykorzystywana do wyboru alternatywnych ustawień interfejsu identyfikowanego przez liczbę w polu *Numer Interfejsu*
5	1	Liczba punktów końcowych (*NumEndpoints*)	Liczba punktów końcowych w tym interfejsie (z wyłączeniem punktu końcowego 0). Wartość pola równa 0 oznacza, że ten interfejs korzysta tylko z punktu końcowego 0
6	1	Klasa interfejsu (*InterfaceClass*)	Kod klasy zgodnie z USB. Wartość 0 oznacza, że ten interfejs nie należy do żadnej z klas USB. Wartość 255 oznacza, że klasa interfejsu jest określona przez producenta
7	1	Podklasa interfejsu (*InterfaceSubClass*)	Kod podklasy zgodnie z USB, związany z klasą określoną w polu *Klasa interfejsu*. Jeżeli *Klasa interfejsu* równa się 0, to wartość w tym polu też musi być 0
8	1	Protokół interfejsu (*InterfaceProtocol*)	Kod protokołu zgodnie ze standardem USB, związany z wartościami na polach *Klasa interfejsu* i *Podklasa interfejsu*. Wartość 0 oznacza, że urządzenie w tym interfejsie nie stosuje protokołu specyficznego dla klasy. Wartość 255 oznacza, że urządzenie w tym interfejsie stosuje protokół producenta.
9	1	Interfejs (*Interface*)	Indeks łańcucha znakowego opisującego dany interfejs

W tabeli 3.4 przedstawiono znaczenie poszczególnych pól deskryptora punktu końcowego. Uzupełnieniem dla wymienionego zestawu deskryptorów są deskryptory łańcuchowe zawierające dodatkowe informacje o urządzeniu w czytelnej dla użytkownika postaci tekstowej.

Tabela 3.4. *Deskryptor punktu końcowego*

Numer pola	Rozmiar pola (liczba bajtów)	Nazwa pola	Znaczenie i wartość
1	1	Długość deskryptora (*DescriptorLength*)	Liczba bajtów w deskryptorze punktu końcowego
2	1	Typ deskryptora (*Descriptor Type*)	Stała określająca typ deskryptora. Wartość 5 oznacza deskryptor punktu końcowego
3	1	Adres punktu końcowego (*EndpointAddress*)	Mapa bitowa określająca adres i własności punktu końcowego: bity 0 – 3 numer punktu końcowego, bity 4 – 6 zarezerwowane, ustawione na 0, bit 7 kierunek (ignorowany w przypadku punktu końcowego kontrolnego): punkt końcowy do zapisu (OUT), punkt końcowy do odczytu (IN).
4	1	Atrybuty (*Attributes*)	Mapa bitowa określająca atrybuty punktu końcowego po konfiguracji urządzenia wartością *Configuration Value*: bity 0 – 1 typ transferu, 00 transfer kontrolny, 01 transfer izochroniczny, 10 transfer masowy, 11 transfer przerwaniowy. Pozostałe bity są zarezerwowane
5	2	Maksymalny rozmiar pakietu (*MaxPacketSize*)	Maksymalny rozmiar pakietu, rozumiany jako liczba bajtów danych w pakiecie, który punkt końcowy może wysłać lub przyjąć
6	1	Odstęp czasowy odpytywania (*Polling interval*)	Odstęp czasowy pomiędzy kolejnymi odpytaniami punktu końcowego wyrażony w ms. W przypadku transferu izochronicznego: 1. W przypadku transferu przerwaniowego: 1 do 255. Pole ignorowane w przypadku transferów masowych i kontrolnych

W tabeli 3.5 przedstawiono znaczenie poszczególnych pól w deskryptorze łańcuchowym.

Tabela 3.5. *Deskryptor łańcuchowy*

Numer pola	Rozmiar pola (liczba bajtów)	Nazwa pola	Znaczenie i wartość
1	1	Długość deskryptora (*DescriptorLength*)	Liczba bajtów w deskryptorze łańcuchowym
2	1	Typ deskryptora (*Descriptor Type*)	Stała określająca typ deskryptora. Wartość 3 oznacza deskryptor łańcuchowy
3	zmienny	Identyfikator języka (*Language Identifier LANGID*) lub Łańcuch (*String*)	W przypadku deskryptora łańcuchowego o numerze 0 pole zawiera kody określające języki, w których dostępny jest łańcuch znakowy (tekst). W przypadku deskryptorów o numerach większych od 0, pole zawiera łańcuch znaków w Unicode stanowiących właściwy tekst

Definicję deskryptorów sformułowałem na podstawie specyfikacji USB 1.x, natomiast interpretację znaczenia ich pól zweryfikowałem, korzystając z książki Dona Andersona *USB System Architecture*.



Specyfikacja uzależnia maksymalny rozmiar pakietu od szybkości transmisji oraz rodzaju transferu.

Dla urządzeń małej szybkości transmisji *MaxPacketSize* równy jest 8 bajtom, a transfery realizowane z takim urządzeniem są ograniczone tylko do kontrolnych i przerwaniowych.

W urządzeniach pełnej i wysokiej szybkości transmisji dopuszczalne są wszystkie 4 rodzaje transferów, a *MaxPacketSize* zależy od wykonania danego punktu końcowego, przy czym nie może być większy od:

- 64 bajtów dla transferu kontrolnego przy pełnej lub wysokiej szybkości,
- 512 bajtów dla transferu masowego przy wysokiej szybkości (high speed),
- 64 bajtów dla transferów masowego i przerwaniowego przy pełnej szybkości (full speed),
- 1024 bajtów dla transferów przerwaniowego i izochronicznego przy wysokiej szybkości,
- 1023 bajtów dla transferu izochronicznego przy pełnej szybkości.

Podane wyżej wartości można odczytać z tabeli 1.3 „Porównanie transferów USB”.

3.2. Deskryptory w urządzeniach zgodnych z USB 2.0

Deskryptory w urządzeniach zgodnych z wersją 2.0 standardu USB tylko nieznacznie różnią się od deskryptorów zgodnych z wersją 1.x. Oczywistą zmianą jest zawartość pola *Wersja USB* w deskryptorze urządzenia (tym razem znajdzie się tam 200 h) i, prawdę mówiąc, w większości przypadków będzie to zmiana jedyna.

Ponadto w wersji 2.0 wyraźnie zaznaczono, że domyślne ustawienie interfejsu urządzenia nie może wymagać pasma dla transferu izochronicznego. Jeżeli urządzenie udostępnia ten rodzaj transferu, musi on być wybrany stosownym ustawieniem interfejsu, z którym związany będzie co najmniej jeden deskryptor punktu końcowego. Niektóre urządzenia USB 1.x spełniają wymienione wymaganie.

Oprócz tego w wersji 2.0 wprowadzono nowe, opcjonalne właściwości izochronicznych punktów końcowych dla pełnej szybkości transmisji. W deskryptorze punktu końcowego, tylko dla izochronicznych punktów końcowych (bity 0 = 0, bit 1 = 1 w polu *Atrybuty*), określono:

- rodzaj synchronizacji (kodowany na bitach 3 i 2 w polu *Atrybuty*),
- rodzaj wykorzystania (kodowany na bitach 5 i 4 w polu *Atrybuty*).

Kolejna zmiana dotyczy odstępu czasowego odpytywania punktu izochronicznego kodowanego na polu *Odstęp czasowy odpytywania* w deskryptorze punktu końcowego. Zamiast 1 ms, odstęp ustawia się w jednostkach:

$$2^{\text{Interval} - 1}\ \text{ms},$$

gdzie Interval jest wartością w polu *Odstęp czasowy odpytywania*.