

# Zmiana długości cyklu izochronicznego w szeregowych interfejsach komunikacyjnych USB 2.0 i IEEE 1394A

Michał Sawicki

## 1. Wprowadzenie

Urządzenia peryferyjne, takie jak mysz, klawiatura czy pendrive, komunikują się z komputerem za pomocą interfejsów komunikacyjnych zgodnie z obowiązującymi protokołami komunikacyjnymi (np. SBP-2, IIDC). Z tego względu kluczowym zadaniem portów jest umożliwienie wykonywania wielu transferów danych różnego typu (np. izochronicznych, asynchronicznych) quasi-jednocześnie na jednej szeregowej magistrali. W pracach [1–3] poruszono problem szeregowania transakcji danych w portach USB i IEEE 1394 zapewniającego realizację różnych transferów danych na jednej magistrali. W porcie USB 2.0 za to zadanie odpowiedzialny jest rozszerzony host kontroler (EHC), a w IEEE 1394a (FireWire) na to zadanie składają się odpowiednie procedury arbitrażowe i mechanizm Interwału Równych Szans.

Asynchroniczne i izochroniczne transfery danych różnią się znacząco pod względem nakładanych na nie wymagań i ograniczeń. W przypadku transferu izochronicznego ważniejsze od bezbłędnego dostarczania danych jest regularne ich przesyłanie, ponieważ w systemach typu A/V (systemy audio-wizyjne) dane multimedialne powinny być transmitowane bez opóźnień, a ewentualne retransmisje danych byłyby niedopuszczalne. Inaczej jest w przypadku transferu asynchronicznego (masowego), gdzie żądane jest wierne przesyłanie danych w możliwie jak najkrótszym czasie, ale nie jest już wymagana regularność dostarczania tych danych. Realizacja obu transferów danych quasi-jednocześnie na jednej magistrali nie jest możliwa w sposób izolowany, tzn. bez wzajemnego wpływu jednego transferu na drugi<sup>1</sup>. Jednym ze skutków tego jest zjawisko zmian długości cyklu izochronicznego, które zostało dokładniej opisane w tym artykule. Zaprezentowano również wyniki obserwacji tego zjawiska w dwóch najpopularniejszych portach USB i IEEE 1394 (FireWire).

Wybrano wersję 2.0 portu USB, ponieważ ta wersja jest obecnie technologią dominującą na rynku interfejsów komunikacyjnych. Wspierana jest przez ogromną liczbę urządzeń na całym świecie. W 2008 roku pojawiła się nowsza wersja (USB 3.0), która jest w pełni kompatybilna z wersją 2.0, ale jej wdrażanie jeszcze potrwa jakiś czas, zanim wszyscy producenci sprzętu komputerowego nie wyposażą swoich produktów w wersję 3.0.

Konkurencyjnym rozwiązaniem dla USB jest port IEEE 1394 (FireWire), którego popularność jest znacznie mniejsza niż wymienionego wcześniej portu USB. Jednakże w szczególnych zastosowaniach jest on „lepszy” od portu USB [6]. Przykładowo wykorzystywany jest w systemach wizyjnych do transmisji obrazu z kamer na urządzenia przechowujące dane multimedialne

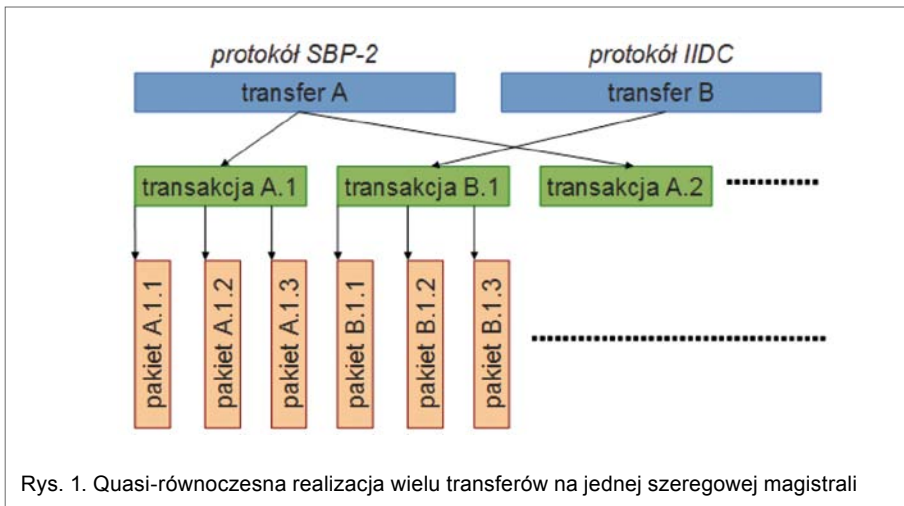
**Streszczenie:** Urządzenia peryferyjne komunikują się z komputerem za pomocą interfejsów komunikacyjnych. Z tego względu kluczowym zadaniem portów jest umożliwienie wykonywania wielu transferów danych różnego rodzaju quasi-jednocześnie na jednej szeregowej magistrali. Taka realizacja różnych transferów danych na jednej magistrali nie jest możliwa w sposób izolowany, tzn. bez wzajemnego wpływu jednego transferu na drugi. Jednym ze skutków tego oddziaływania jest zjawisko zmian długości cyklu izochronicznego, które zostało dokładniej opisane w tym artykule. Zaprezentowano również wyniki obserwacji tego zjawiska w dwóch najpopularniejszych portach USB i IEEE 1394 (FireWire).

Eksperymenty pokazały, że w przypadku portu USB transmisja masowa może wpływać na transmisję izochroniczną, skracając długość mikroramki, a co za tym idzie, ograniczając pasmo w mikroramce. Rozwiązania wykorzystujące interfejs FireWire muszą uwzględniać zjawisko drgań cyklu izochronicznego. Skutkuje to zwiększeniem rozmiaru i wzrostem złożoności czasowej oprogramowania (np. sterowników kamer FireWire).

**Abstract:** *Peripheral devices interact with computer via the communication interfaces. Therefore a key task is to allow the ports perform many different types of data transfers quasi-simultaneously on one serial bus. Execution of different data transfers on the same bus is not possible without the interaction of these transfers. One of the consequences of this interaction is a length change of the isochronous cycle (cycle jitter), which is more fully described in this article. This paper presents the results of the observation of this phenomenon in the two most popular ports: USB and IEEE 1394 (FireWire).*

*Experiments have shown that the bulk USB transmission may interact with the isochronous transmission by shortening the length of microframe and reducing the bandwidth within microframe. Solutions based on the FireWire port must reduce the effects of cycle jitter, so the size and complexity of this software is greater (e.g. FireWire camera driver).*

(pamięci masowe, stanowiska komputerowe). W celu porównania dwóch portów ze względu na oddziaływanie transferów danych na siebie wybrano wersję IEEE 1394a, która znajduje się w tej samej klasie szybkości transmisji co USB 2.0 (w USB 2.0 szybkość 480 Mb/s, IEEE 1394a 400 Mb/s).



Rys. 1. Quasi-równoczesna realizacja wielu transferów na jednej szeregowej magistrali

## 2. Transmisja danych w szeregowych interfejsach komunikacyjnych

W standardzie USB wyszczególniono cztery typy transferów [8]: kontrolny, przerwaniowy, izochroniczny i masowy. Pierwszy z nich wykorzystywany jest do instalacji i konfiguracji podłączanych urządzeń do magistrali. W przypadku transferów przerwaniowych i izochronicznych położono nacisk na regularne dostarczanie danych, przy czym w przerwaniowym wymagane jest również bezbłędne dostarczanie danych, co może wiązać się z ewentualnymi retransmisjami uszkodzonych informacji. Transfer przerwaniowy umożliwia przesyłanie niewielkich ilości danych w regularnych odstępach czasu i jest wykorzystywany do komunikacji z urządzeniami klasy HID (*Human Interface Device*), przykładowo do komunikacji z myszką lub klawiaturą. Transfer izochroniczny umożliwia natomiast regularne przesyłanie dużej ilości danych. Jednakże nie są możliwe jakiegokolwiek retransmisje, ze względu

na brak w tym przypadku mechanizmu potwierdzania danych. Stosowany jest w transmisji danych A/V, na przykład przy transmisji obrazu z kamery internetowej. Ostatni z transferów, masowy, gwarantuje dostarczanie bezbłędnych danych (mogą wystąpić retransmisje) w możliwie najkrótszym czasie zgodnie z zasadą „usilnych starań” [8–9]. W tym przypadku nie jest zapewnione regularne przesyłanie danych. Wykorzystywany jest do transmisji danych z/do pamięci masowych, np. pendrive, zewnętrzne dyski twarde.

W interfejsie FireWire dostępne są dwa typy transferów danych [9]: izochroniczny i asynchroniczny, które są „naturalnymi” odpowiednikami transferów USB odpowiednio izochronicznego i masowego. Różnią się na poziomie budowy transakcji i struktury pakietów. Ze względu na duże podobieństwo tych transferów w dalszej części będą one omawiane razem.

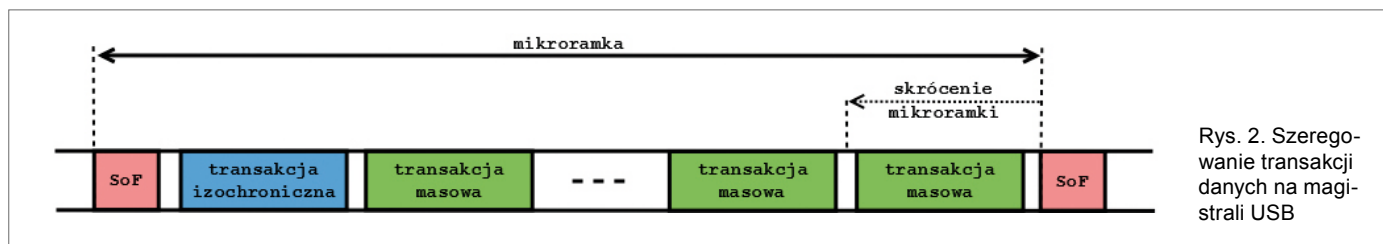
Transfery danych podzielone są na mniejsze jednostki: transakcje, które

z kolei składają się z pakietów. Taki podział (rys. 1) umożliwia quasi-równoczesną realizację wielu transferów danych na jednej magistrali za pomocą odpowiedniego mechanizmu szeregowania transakcji danych. W dalszej części omówione zostaną tylko transfery izochroniczny i asynchroniczny, ponieważ w obu przypadkach przesyłane są duże ilości danych, a dodatkowo na transfer izochroniczny nałożone zostało wymaganie regularnego dostarczania tych danych.

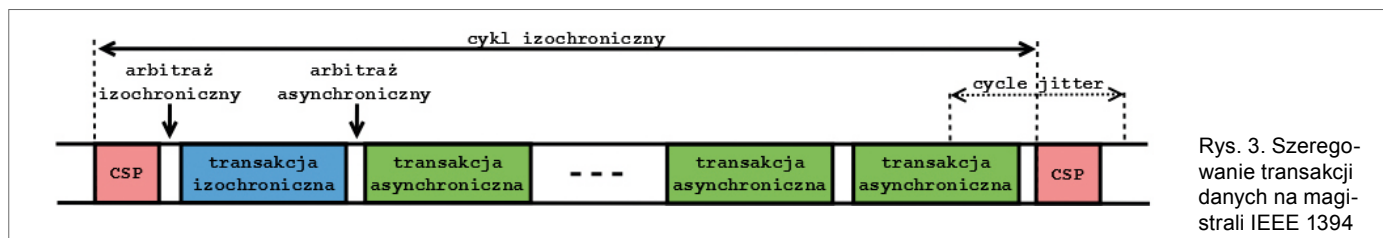
### 2.1. Transakcja izochroniczna

Transfer izochroniczny umożliwia przesyłanie dużych ilości danych w regularnych odstępach czasu. Nie jest gwarantowane bezbłędne dostarczenie danych, w związku z czym nie są wykonywane retransmisje uszkodzonych informacji, mimo że jest weryfikowana poprawność pakietów za pomocą sum kontrolnych CRC. Głównym zastosowaniem tego transferu jest przesył danych o krótkim „czasie życia”, przykładowo danych multimedialnych.

W obu interfejsach pasmo podzielone jest na cykle izochroniczne (w USB nazywane mikroramkami) o nominalnej długości 125  $\mu$ s. W ramach cyklu realizowane są pojedyncze transakcje zleconych transferów izochronicznych. W USB możliwa jest tzw. szerokopasmowa transmisja izochroniczna, umożliwiająca realizację wielu transakcji tego samego transferu w jednej mikroramce, ale jest to przypadek szczególny, dalej nierozpatrywany. Na początku cyklu przesyłany jest odpowiednio pakiet początku cyklu (mikroramki). Po tym pakiecie realizowane są transakcje, przy czym kolejność ustalana jest w przypadku USB przez EHC (rys. 2), a w FireWire w wyniku arbitrażu izochronicznego i asynchronicznego (rys. 3) oraz na podstawie zasady



Rys. 2. Szeregowanie transakcji danych na magistrali USB



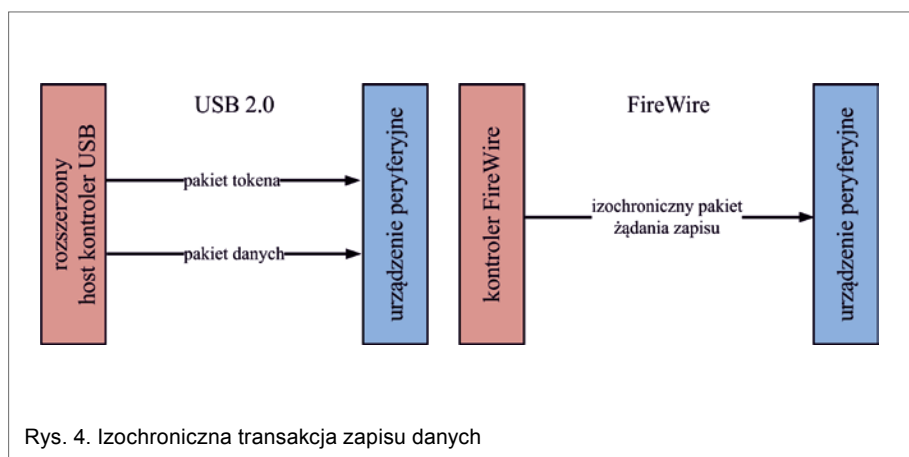
Rys. 3. Szeregowanie transakcji danych na magistrali IEEE 1394

Interwału Równych Szans. Wymienione mechanizmy i sposoby szeregowania transakcji danych zostały dokładniej opisane w publikacjach [5,8–9].

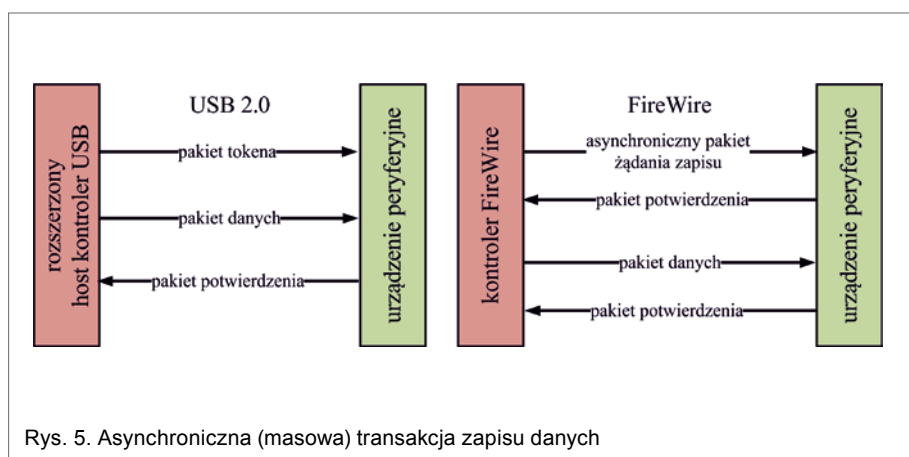
W porcie USB dane izochroniczne mogą być zapisywane, jak również odczytywane z urządzenia. Transakcje izochroniczne USB (rys. 4) zaczynają się od pakietu tokena, w którym zawarte są parametry transakcji, m.in. rodzaj transakcji, rozmiar przesyłanych danych, adres odbiorcy itd. Następnie przesyłany jest pakiet danych zawierający dane, np. multimedialne. W tego typu transakcjach nie występuje mechanizm potwierdzeń, przez co pakiet danych kończy transakcję. W FireWire możliwa jest jedynie operacja zapisu danych izochronicznych i w tym przypadku transakcja (rys. 4) składa się tylko z jednego izochronicznego pakietu żądania zapisu danych. Zaletą takiego rozwiązania jest możliwość wykonywania operacji rozgłoszenia danych, przesyłając tylko jeden pakiet izochroniczny.

## 2.2. Transakcja asynchroniczna

Transfer asynchroniczny (masowy) gwarantuje dostarczanie poprawnych danych, przez co transakcje takiego transferu muszą być rozbudowane o mechanizm potwierdzeń. W przypadku transakcji masowej USB (rys. 5) oprócz pakietu tokena oraz pakietu danych przesyłany jest również pakiet potwierdzenia. W przypadku negatywnej odpowiedzi wykonywana jest retransmisja uszkodzonych danych. W obu interfejsach możliwe są operacje odczytu i zapisu danych, a dodatkowo w FireWire możliwa jest jeszcze operacja blokowania polegająca



Rys. 4. Izochroniczna transakcja zapisu danych



Rys. 5. Asynchroniczna (masowa) transakcja zapisu danych

na odczytaniu, zmodyfikowaniu i zapisaniu danych w ramach jednej transakcji.

Transakcja asynchroniczna FireWire (rys. 5) składa się z dwóch faz (subakcji). W ramach pierwszej fazy przesyłane są asynchroniczny pakiet żądania odczytu lub zapisu danych oraz pakiet potwierdzenia, przy czym potwierdzenie to dotyczy tylko fazy, a nie całej transakcji.

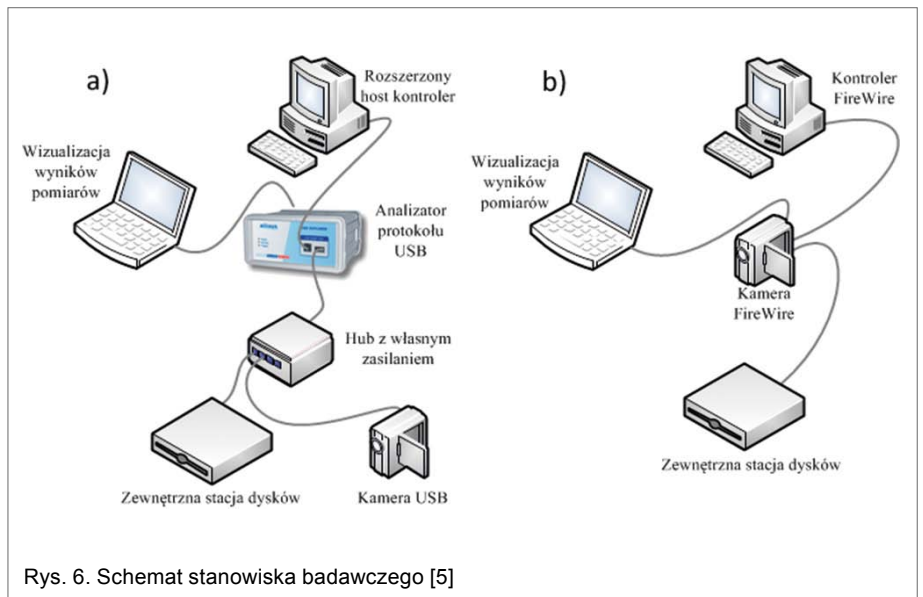
Pakiet żądania, podobnie jak pakiet tokena, specyfikuje rodzaj operacji i jej parametry. W drugiej fazie przesyłane są pakiety odpowiedzi i potwierdzenia tej odpowiedzi. W tym przypadku również pakiet potwierdzenia dotyczy tylko tej fazy. Pakiet odpowiedzi zawiera odczytane dane dla transakcji odczytu lub potwierdzenie poprawności zapisu danych,

które zostały przesłane w pakiecie żądania zapisu w przypadku transakcji zapisu. Transakcja blokowania łączy w sobie obie operacje zapisu i odczytu, przez co dane służące do modyfikacji przesyłane są w pakiecie żądania, a pakietem odpowiedzi przesyłane są dane odczytane przed modyfikacją.

### 3. Zjawisko zmiany długości cyklu izochronicznego

W porcie USB możliwe jest skrócenie lub wydłużenie mikroramki maksymalnie o 1  $\mu$ s, modyfikując zawartość jednego z rejestrów EHC [7]. Zmiana długości mikroramki wykonywana jest w celu zsynchronizowania nadajników i odbiorników po stronie EHC i urządzenia peryferyjnego.

Dostęp do łącza udzielany jest urządzeniom USB przez EHC i to on nadzoruje wykonanie transakcji, pilnując, aby każda z nich została zrealizowana w ramach jednej mikroramki oraz aby nie doszło do sytuacji, kiedy pod koniec mikroramki, w momencie kiedy powinien zostać nadany pakiet SoF, nadal realizowana jest niezakończona transakcja. Inaczej jest w porcie FireWire, gdzie o dostęp do łącza urządzenia rywalizują w ramach procedury arbitrażowej. Może dojść do sytuacji, kiedy pod koniec cyklu izochronicznego rozpoczęta zostanie jedna z faz transakcji asynchronicznej. W takiej sytuacji nie wolno przerywać transmisji danych, tylko następuje przesunięcie momentu nadania pakietu CSP (rys. 3). Oznacza to, że kolejny cykl rozpoczyna się później, niż było to planowane. Urządzenie, które znajduje się na szczycie drzewiastej topologii systemu FireWire, pełni rolę korzenia, do zadań którego na-



Rys. 6. Schemat stanowiska badawczego [5]

leży m.in. kontrola cyklu. W momencie, kiedy rozpoczęcie cyklu jest przesunięte w czasie, przez przedłużającą się ostatnią transakcję asynchroniczną w poprzednim cyklu, zadaniem kontrolera cyklu (czyli korzenia) jest powrót do nominalnego początku cyklu. Następuje skrócenie przesuniętego w czasie cyklu o czas, jaki został poświęcony na dokończenie przedłużającej się transakcji. Cała operacja skutkuje utratą regularności dostarczania danych i nazywana jest zjawiskiem drgań cyklu izochronicznego (ang. *cycle jitter*). Najpierw cykl jest wydłużony przez przedłużającą się transakcję, a potem skracany przez kontroler cyklu w celu powrotu do nominalnego momentu początku. Zjawisko to można zniwelować, buforując izochroniczne dane. W dalszej części artykułu zaprezentowano wyni-

ki obserwacji tych zjawisk dla systemu USB (obserwacja skrócenia mikroramki) i FireWire (zjawisko drgań cyklu izochronicznego).

### 4. Pomiar długości cyklu izochronicznego

Zestawiono cztery stanowiska badawcze: dwa z systemem USB i dwa z FireWire. W pierwszym przypadku (rys. 6 a) do komputera podłączono poprzez analizator interfejsu USB urządzenia peryferyjne, takie jak kamera internetowa i zewnętrzna stacja dysków. Dodatkowo należało włączyć między analizator a urządzenia peryferyjne huba z własnym zasilaniem, ponieważ sam analizator nie posiada własnego zasilania i nie byłby w stanie odpowiednio zasi-

reklama



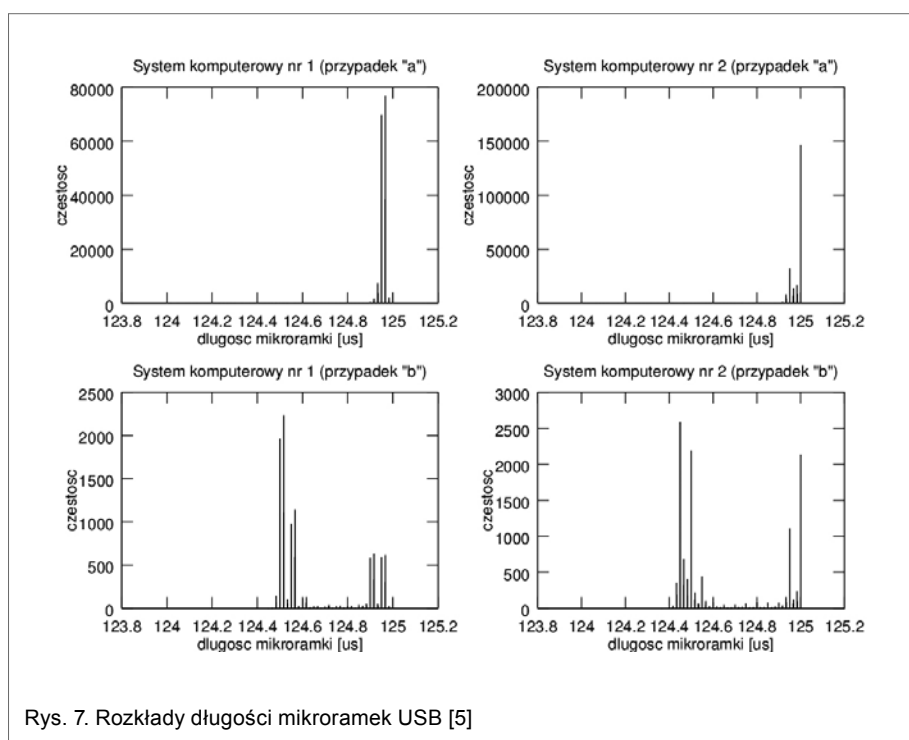
lić tych urządzeń peryferyjnych. Jako pamięć masową wybrano stację dysków wyposażoną wewnątrz w interfejs SATA, a nie np. pendrive'a, ponieważ do komunikacji z pendrivem wyposażonym w pamięć Flash wykorzystywany jest dodatkowo protokół PING-NYET, służący do kontroli gotowości tej pamięci. Pakiety przesyłane w ramach tego protokołu mogłyby wpływać na uzyskane wyniki, dlatego zdecydowano wybrać dysk twardy, który nie wykorzystuje tego protokołu. Oprócz komputera, który pełnił rolę EHC, podłączono do analizatora jeszcze jeden komputer służący do wizualizacji otrzymanych pomiarów z analizatora. Metodę pomiaru długości mikroramek zaprezentowano w kolejnym punkcie.

W przypadku stanowisk z portami FireWire do komputera podłączono szeregowo kamerę internetową i zewnętrzną stację dysków poprzez interfejs FireWire. Dodatkowo do kamery podłączono komputer odpowiedzialny za akwizycję i wizualizację pomiarów, ponieważ proces rejestracji długości cykli nie był realizowany za pomocą analizatora [4], tak jak to miało miejsce w przypadku USB, tylko za pomocą oprogramowania, które stworzono dla celów tego badania i uruchomiono na komputerze podłączonym do kamery.

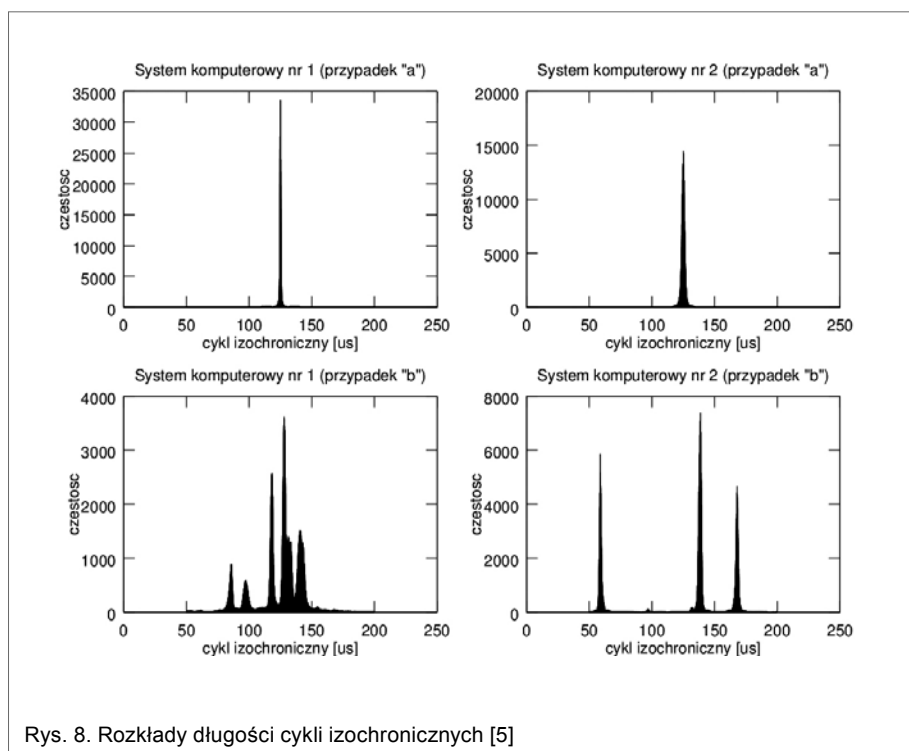
#### 4.1. Metoda pomiaru długości mikroramki USB

Pomiary długości mikroramki wykonano na obu stanowiskach z USB dla dwóch przypadków (rys. 7 „a” i „b”), gdy pobierano tylko obraz z kamery (transfer izochroniczny) oraz gdy dodatkowo odczytywano dane z zewnętrznej stacji dysków (transfer asynchroniczny). Dla obu przypadków i dwóch stanowisk wyposażonych w system komunikacyjny USB sporządzono histogramy przedstawione na rys. 7. Za pomocą analizatora USB, który wpięty był pomiędzy EHC a urządzenie peryferyjne, rejestrowano moment wysłania pakietu SoF. Wykonując różnicę dwóch czasów dla kolejnych pakietów SoF, uzyskiwano długości mikroramek.

Jeżeli realizowany był tylko transfer izochroniczny (przypadek „a”), to długości mikroramek skupiały się wokół wartości nominalnej, czyli 125  $\mu$ s, tzn. mikroramki nie były ani skracane, ani wydłużane w celu utrzymania synchronizacji pomiędzy EHC a urządzeniem peryferyjnym. Jeśli dodatkowo odczytywano w tym samym czasie informacje



Rys. 7. Rozkłady długości mikroramek USB [5]



Rys. 8. Rozkłady długości cykli izochronicznych [5]

z dysku twardego (czyli realizowany był transfer masowy, przypadek „b”), to następowało skracanie większość mikroramek o ok. 0,5  $\mu$ s, co skutkowało utratą regularności i zwiększeniem szybkości dostarczania kolejnych porcji danych (średnio częściej wykonywane były transakcje izochroniczne).

#### 4.2. Metoda pomiaru długości cyklu izochronicznego IEEE 1394

Podobny eksperyment przeprowadzono dla systemu komunikacyjnego FireWire, przy czym w tym przypadku pomiar długości cyklu izochronicznego odbywał się na dodatkowym komputerze bez udziału analizatora interfejsu.

Na tym komputerze rejestrowano moment wygrania arbitrażu izochronicznego (czyli uzyskania dostępu do łącza), co zdarzało się raz na cykl izochroniczny. Obliczając różnicę pomiędzy dwoma czasami wygrania arbitrażu w sąsiednich cyklach izochronicznych, uzyskiwano długość cyklu. Na rys. 8 zaprezentowano histogramy otrzymanych pomiarów długości cyklu izochronicznego. W przypadku, gdy przesyłano jedynie obraz z kamery (przypadek „a”), długości mikroramek skupiały się podobnie jak dla USB wokół wartości nominalnej, czyli 125  $\mu$ s. Natomiast jeżeli dodatkowo odczytywano jeszcze pliki z dysku twardego (przypadek „b”), to otrzymany rozkład jest już wielomodowy, co wskazuje jednoznacznie na występowanie zjawiska drgań cyklu izochronicznego opisanego we wcześniejszym punkcie. Cykl izochroniczny mógł być skrócony nawet o 50–60  $\mu$ s, co skutkowało utratą regularności dostarczania danych.

## 5. Podsumowanie

W przedstawionym artykule zaprezentowano wyniki obserwacji zmian długości cyklu izochronicznego (mikroramki) dla dwóch najpopularniejszych interfejsów komunikacyjnych USB i FireWire. Eksperymenty pokazały, że w przypadku portu USB transmisja masowa może wpływać na transmisję izochroniczną, skracając długość mikroramki, a tym samym zwiększając częstotliwość dostarczania danych izochronicznych (o co najwyżej 65 Hz). Przy czym samo skrócenie mikroramki nie wpływa znacząco na utratę regularności dostarczania danych, jednak z drugiej strony ogranicza pasmo (jednej mikroramki), w ramach którego można zrealizować jeszcze jedną transakcję. Rozwiązania wykorzystujące interfejs FireWire muszą uwzględniać zjawisko drgań cyklu izochronicznego, co skutkuje m.in. zwiększeniem rozmiaru i wzrostem złożoności czasowej oprogramowania (np. sterowników kamer FireWire). Podsumowując, interfejs USB zapewnia regularne dostarczanie danych izochronicznych, nawet w przypadku, gdy wykonywany jest quasi-równocześnie inny transfer danych. W porcie FireWire moment rozpoczęcia transakcji izochronicznej, a tym samym moment przesłania kolejnych danych izochronicznych zależy od liczby i długości realizowanych transakcji asynchronicznych i wymaga od

powiedniej obsługi w oprogramowaniu urządzenia.

## Przypisy

- 1 Problem ten został rozwiązany w interfejsie Thunderbolt. W przewodzie tego interfejsu producent przewidział dwa osobne kanały do transmisji strumieni danych i audio.

## Literatura

- [1] HUANG CH-Y., KUO T-W., PANG A-CH.: *QoS Support for USB 2.0 Periodic and Sporadic Device Requests*. IEEE International Real-Time Systems Symposium, 2004.
- [2] HUANG CH-Y., KUO T-W., PANG A-CH.: *QoS Storage Subsystems using IEEE 1394*. ACM Transactions on Storage 2009
- [3] OTTEN F., FOSS R.: *Ensuring QoS in Professional Audio Networks which use IEEE 1394b SATNAC 2010 Conference*
- [4] STEINBERG D., BIRK Y.: *An empirical analysis of the IEEE-1394 serial bus protocol*, IEEE Micro, Luty 2000.
- [5] SAWICKI M., MAĆKOWSKI M.: *Wpływ transmisji asynchronicznej na transfer danych multimedialnych w szeregowych interfejsach komunikacyjnych*. Międzynarodowa Konferencja „Naukowa Computer Networks 2013”, Studia Informatica, vol. 34, no 3 (113), Lwówek Śląski 2013.
- [6] SAWICKI M.: *Porównanie interfejsów USB i IEEE 1394a na podstawie czasu realizacji asynchronicznego transferu danych*. „Napędy i Sterowanie” 11/2012.
- [7] Specyfikacja Enhanced Host Controller Interface for Universal Serial Bus Ver.1.0-2002.
- [8] MIELCZAREK W.: *USB. Uniwersalny interfejs szeregowy*, Helion, Gliwice 2005.
- [9] MIELCZAREK W.: *Szeregowy interfejs cyfrowy FireWire*. Wydawnictwo Politechniki Śląskiej, Gliwice 2010.

mgr inż. Michał Sawicki – Instytut Informatyki Politechniki Śląskiej

artykuł recenzowany