

Łukasz HERB
Politechnika Śląska, Instytut Informatyki

ANALIZA WPŁYWU ARCHITEKTURY WEWNĘTRZNEJ KONCENTRATORA DANYCH PROTOKOŁU MODBUS RTU NA CZAS TRWANIA CYKLU SIECI

Streszczenie. W niniejszym artykule przedstawiono wpływ architektury sprzętowej koncentratorów danych protokołu Modbus na czas trwania cyklu sieci w zaproponowanych modelach systemów. Oszacowanie parametrów sieci dla systemu o topologii magistrali oraz systemu z koncentratorem umożliwiło wskazanie zależności wymaganych zasobów sprzętowych od liczby węzłów w sieci, dla których pojawia się zysk, wynikający ze stosowania koncentracji.

Słowa kluczowe: Modbus, RS-485, koncentrator, Omnet++

ANALYSIS OF INFLUENCE OF THE MODBUS RTU DATA HUB HARDWARE ARCHITECTURE ON THE NETWORK CYCLE TIME

Summary. This paper presents an influence of the Modbus data hub's hardware architecture on network cycle time in proposed systems models. Features of the Modbus protocol, network microprocessor devices and a way to simulate it in Omnet++ software are presented. Estimated parameters of the network for system in bus topology and system with the hub allowed to show the dependency between required hardware configuration and the number of nodes in the network that provide profits from using the hub.

Keywords: Modbus, RS-485, hub, Omnet++

1. Wprowadzenie

Jednym z pierwszych zadań podczas projektowania systemu sterowania jest określenie topologii sieci. Należy wówczas wziąć pod uwagę m.in. liczbę węzłów, których dane mają być wymieniane, ograniczenia czasowe dotyczące cyklu sieci oraz dostępne urządzenia i pro-

tokoły przez nie obsługiwane. Jeżeli zadanie takie postawione zostanie przed dostawcą elementów sieciowych, możliwe jest wykonanie urządzeń o architekturze sprzętowej dedykowanej określonej topologii sieci.

W tym artykule przedstawiony zostanie wpływ architektury sprzętowej koncentratorów danych protokołu Modbus, komunikujących się z urządzeniami podrzędnymi za pośrednictwem magistrali RS-485, na długość cyklu sieci. Przedstawiona zostanie wymagana liczba dostępnych w urządzeniu portów szeregowych, aby cykl sieci był krótszy niż w przypadku topologii magistrali.

W celu porównania parametrów sieci z koncentracją oraz topologii magistrali zaproponowano ich cyfrowe modele oraz przeprowadzono symulacje w środowisku Omnet++.

2. Koncentrator danych protokołu Modbus

Koncentrator jest elementem sieciowym, pośredniczącym w komunikacji pomiędzy węzłami podrzędnymi, których liczba uzależniona jest od liczby jego portów dolnych, a systemem nadrzędnym komunikującym się z portem górnym. Ponieważ zastosowano protokół Modbus, wymiana danych polega na odczycie oraz zapisie rejestrów pamięci.

2.1. Koncentrator jako urządzenie sieciowe

Konfiguracja tego urządzenia odbywa się poprzez zapis scenariusza wymian, opisanego parametrami:

- numer portu dolnego, do którego podłączono urządzenie,
- kierunek (zapis/odczyt),
- liczba rejestrów, których dotyczy wymiana,
- adres pierwszego z wymienianych rejestrów w pamięci urządzenia podrzędnego,
- adres pierwszego z wymienianych rejestrów w pamięci koncentratora.

Zapisany scenariusz wymian realizowany jest w nieskończonej pętli dokonując wymiany danych pomiędzy podłączonymi węzłami a własną pamięcią koncentratora. Odczyt pamięci możliwy jest w każdej chwili poprzez port górny, co umożliwia dostęp do aktualnie przechowywanego stanu podłączonych urządzeń.

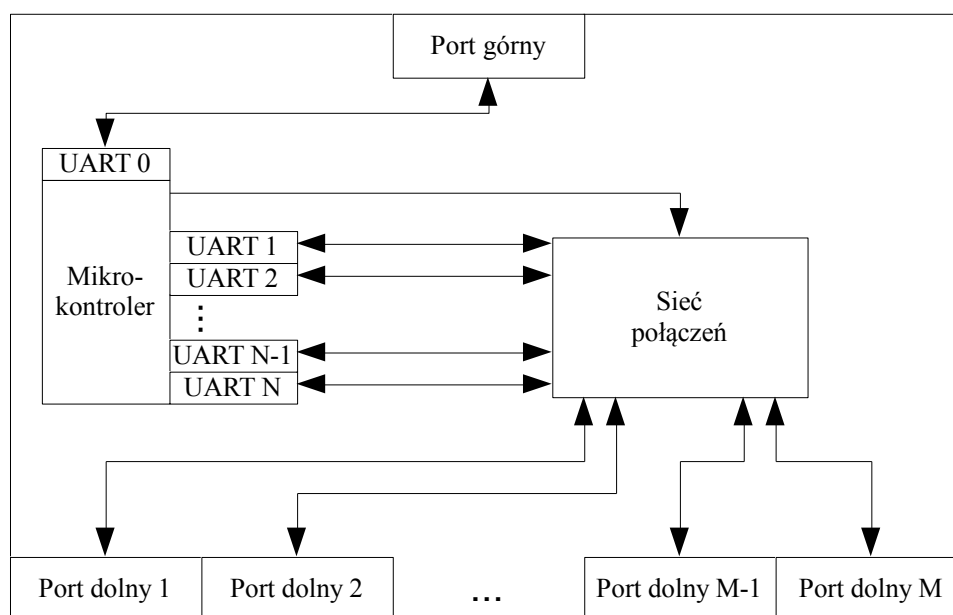
Zastosowanie koncentratora eliminuje konieczność kolejnego odpytywania wszystkich węzłów, co ma miejsce w topologii magistrali. Ponieważ rejestry urządzeń podłączonych do koncentratora mapowane są do wskazanych rejestrów w jego pamięci, możliwe jest jednoczesne odczytanie wszystkich wymienionych danych. Należy zwrócić uwagę, że są to dane aktualnie przechowywane w pamięci koncentratora, a ich maksymalny wiek odpowiada długości

cyklu sieci. Są zatem starsze niż aktualnie znajdujące się w urządzeniu slave o czas realizacji całego scenariusza wymian przez koncentrator.

W analizowanym przypadku wszystkie węzły w sieci połączone są z portami dolnymi koncentratora, zatem czas cyklu sieci jest równy czasowi realizacji scenariusza wymian.

2.2. Koncentrator jako system mikroprocesorowy

Czas realizacji scenariusza wymian jest silnie związany z budową wewnętrzną urządzenia mikroprocesorowego, jakim jest koncentrator, oraz zaimplementowanymi w nim algorytmami komunikacji. Na rys. 1 przedstawiono elementy wchodzące w skład tego systemu. Wyróżnić należy dwa bloki wewnętrzne: mikrokontroler i sieć połączeń oraz zewnętrzne: porty dolne oraz port górny.



Rys. 1. Schemat blokowy budowy wewnętrznej koncentratora

Fig. 1. Block diagram of the hub's internal design

Mikrokontroler jest elementem elektronicznym zbudowanym z rdzenia mikroprocesora odpowiedzialnego za realizację zapisanych w pamięci algorytmów oraz układów peryferyjnych, takich jak porty szeregowy, pamięć programu i danych itp. To przede wszystkim od ilości portów szeregowych (ang. UART – *Universal Asynchronous Receiver and Transmitter*) zależy będzie algorytm komunikacji na portach dolnych.

Porty dolne oraz port górny są wyprowadzeniami sygnałów komunikacji na zewnątrz urządzenia. W opisywanym przypadku są to interfejsy magistrali RS-485 wykorzystujące scalone sterowniki.

Sieć połączeń realizuje komunikację pomiędzy mikrokontrolerem a portami dolnymi. W najprostszym przypadku liczba portów dolnych jest mniejsza bądź równa liczbie kontrol-

rów UART (nie wliczając kontrolera połączonego z portem górnym), co pozwala na ich bezpośrednie połączenie. W przypadku zastosowania większej liczby portów dolnych sieć połączeń odpowiada za przełączanie komunikacji pomiędzy portami szeregowymi a portami dolnymi tak, by przez każdy z nich nawiązana została komunikacja. Algorytm przełączania realizowany jest przez mikrokontroler, który sygnałami sterującymi wpływa na pracę sieci połączeń. W roli sieci połączeń stosowane mogą być układy logiki programowalnej CPLD lub bogatsze w zasoby FPGA, które dodatkowo pozwolą na realizację kolejnych kontrolerów UART w swojej strukturze.

2.3. Algorytm przełączania komunikacji

Aby możliwa była realizacja scenariusza wymian, w którym zdefiniowano większą liczbę portów dolnych niż dostępna pula kontrolerów UART, zaproponowano algorytm przełączania połączeń. Przez N oznaczono dostępną liczbę kontrolerów portu szeregowego, a przez M liczbę urządzeń podrzędnych, z którymi koncentrator wymienia dane. Mikroprocesor realizuje następujące kroki:

- 1) połącz pierwsze N portów dolnych z kontrolerami UART,
- 2) wyślij żądanie przez każdy z kontrolerów,
- 3) czekaj przez maksymalny możliwy czas trwania najdłuższej z przeprowadzanych aktualnie transakcji,
- 4) przełącz sieć połączeń według wzoru (1),
- 5) skocz do punktu 2.

$$P_i = N(t-1) \bmod M + i, \quad t = 1, 2, \dots \quad (1)$$

gdzie: i to numer portu szeregowego, natomiast P_i to odpowiadający mu numer portu dolnego, t – numer kolejnej iteracji algorytmu, N – liczba portów UART, M – liczba portów dolnych.

Zauważyć można, że topologia magistrali odpowiada działaniu koncentratora o jednym porcie UART.

3. Symulacja działania sieci

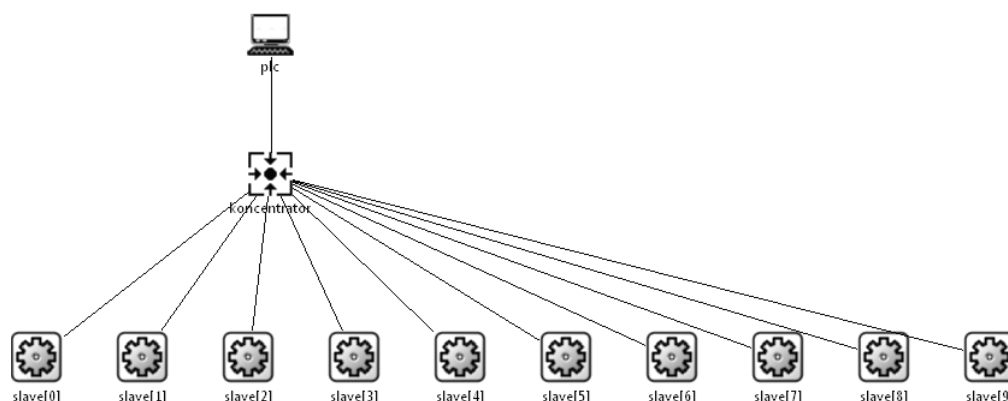
W celu zbadania zachowania sieci zbudowanej z użyciem koncentratorów oraz porównania wyników z siecią o topologii magistrali zbudowano ich cyfrowe modele. Następnie przeprowadzono symulacje wymiany komunikatów w sieciach z różnymi liczbami abonentów oraz różnymi konfiguracjami sprzętowymi koncentratorów.

3.1. Cyfrowy model sieci o protokole Modbus RTU

Protokół Modbus opublikowany został przez firmę Modicon w 1979 roku [1]. Ze względu na otwarty standard i łatwość w implementacji jest on nadal szeroko stosowany w sieciach przemysłowych [3]. Dla potrzeb symulacji przyjęto, że wykorzystywany jest protokół Modbus RTU, a implementowane są jego dwie funkcje:

- odczyt wielu rejestrów (ang. *Read Holding Registers*) kod funkcji 3_h; długość ramki żądania: 8B; długość ramki odpowiedzi: 5B + 2B na każdy odczytywany rejestr;
- zapis wielu rejestrów (ang. *Write Multiple register*) kod funkcji 10_h; długość ramki żądania: 9B + 2B na każdy zapisywany rejestr; długość ramki odpowiedzi: 8B.

Założono dodatkowo, że scenariusz wymian planuje w pierwszej kolejności odczyt danych ze wszystkich urządzeń, a następnie zapis do nich w tej samej kolejności. Komunikacja odbywa się z wykorzystaniem portu szeregowego z prędkością 115200 baud. Znak zbudowany jest z 8 bitów danych i jednego bitu stop, nie zastosowano bitu parzystości.



Rys. 2. Model sieci w programie Omnet++

Fig. 2. Network model in Omnet++ software

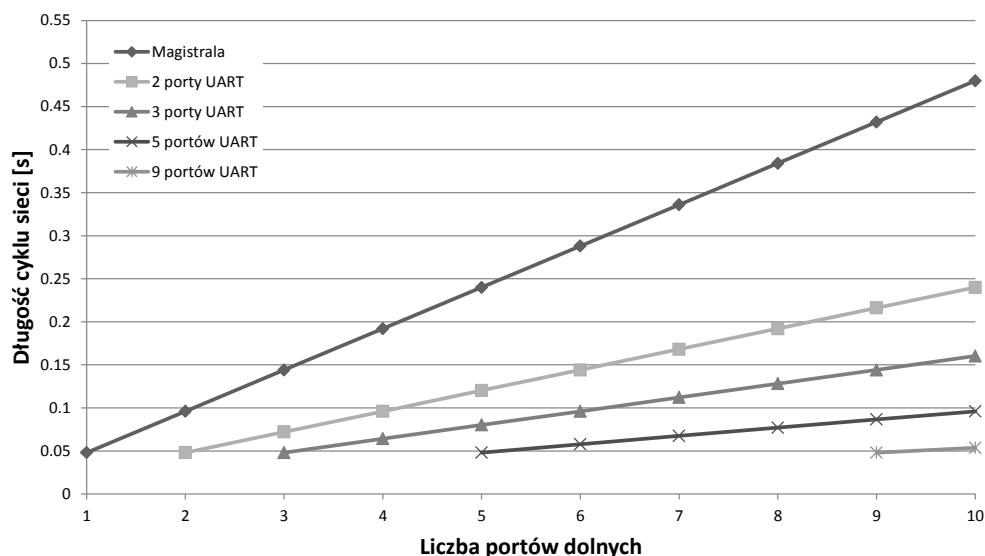
Ponieważ sieć pracuje stabilnie [2], mierzony czas trwania jej cyklu zdefiniowano jako czas pomiędzy kolejnymi odpowiedziami pierwszego podłączonego węzła.

Na rys. 2 przedstawiono graficzny model sieci, wygenerowany przez program Omnet++. Zawiera on modele koncentratora i urządzeń przez niego odpytywanych (w środowisku symulacyjnym przedstawione jako moduły typu *simple*) oraz magistral łączących (kanały komunikacyjne rozszerzające funkcjonalność klasy *DatarateChannel*), zbudowane według wcześniej wymienionych założeń. Moduły umieszczono w elemencie nadrzędnym (*networks*), który je parametryzuje oraz przechowuje informację o połączeniach między nimi.

W celu symulacji układu zdarzeń dyskretnych, jakim jest przedstawiony model sieci, wykorzystano koncepcję planowania zdarzeń (zrealizowaną z użyciem metody *handleMessage*, implementowanej przez każdą z klas reprezentujących moduły). Jako zdarzenie przyjęto moment wysłania lub otrzymania komunikatu przez dowolny moduł, a także upływanie czasu oczekiwania na odpowiedź urządzenia.

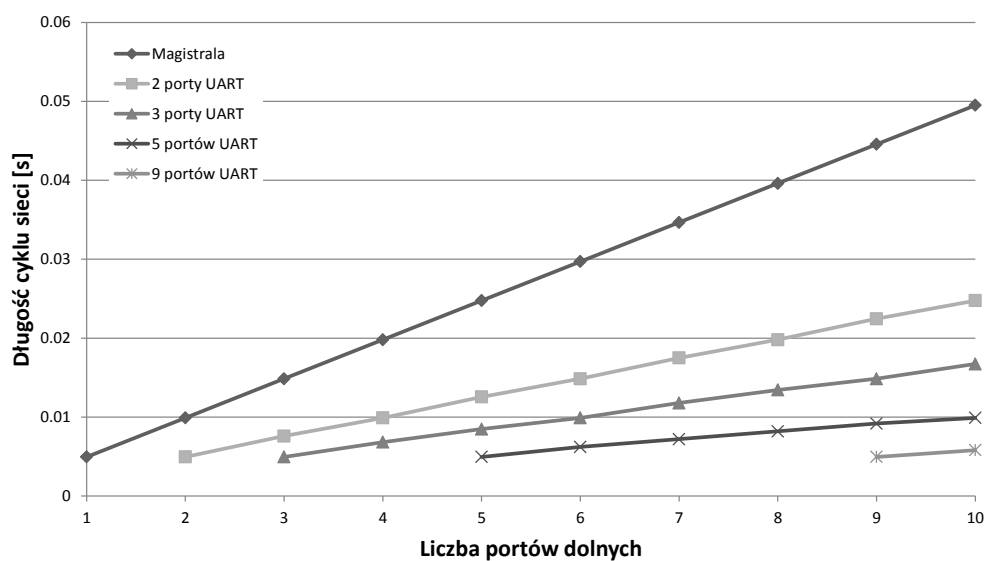
3.2. Przeprowadzone eksperymenty

Proponowany system, wykorzystujący koncentratory porównano z siecią o topologii magistrali. Symulację magistrali przeprowadzono ograniczając liczbę portów szeregowych dostępnych w mikrokontrolerze koncentratora do jednego, liczba portów dolnych odpowiada natomiast liczbie podłączonych urządzeń slave. Konfiguracja taka powoduje kolejne odpytywanie węzłów, jak ma to miejsce w przypadku ich obecności na jednej magistrali.



Rys. 3. Średnie wartości długości cyklu sieci dla różnej liczby portów dolnych. Maksymalne długości ramek

Fig. 3. Mean value of the network cycle for different number of lower ports. Maximum length of frames



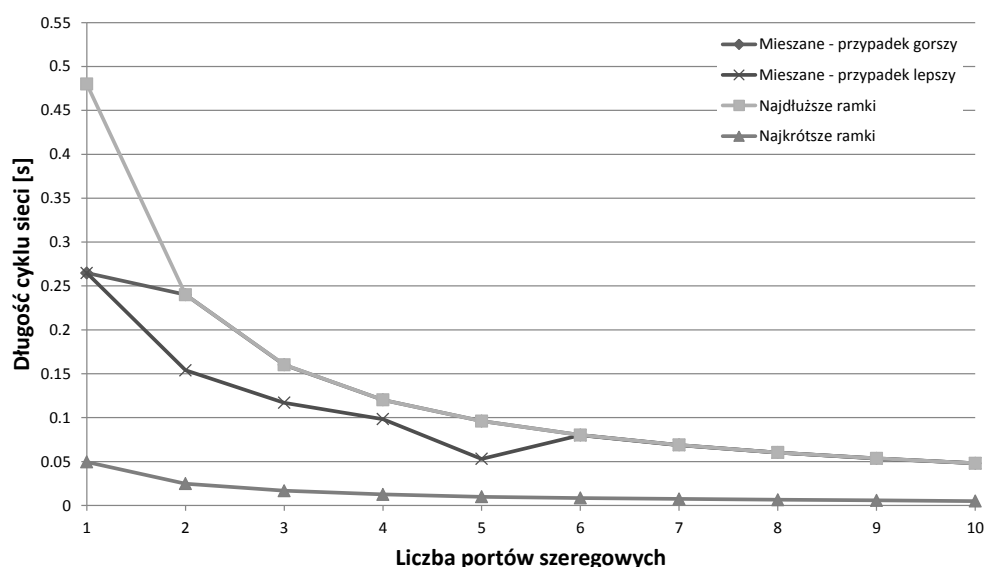
Rys. 4. Średnie wartości długości cyklu sieci dla różnej liczby portów dolnych. Minimalne długości ramek

Fig. 4. Mean value of the network cycle for different number of lower ports. Minimum length of frames

Czas eksperymentu ustalono na jedną godzinę, co w przypadku najdłuższych możliwych ramek i wybraniu topologii magistrali umożliwia wykonanie 7500 cykli sieci. Zbadano dwa skrajne przypadki: odczyt i zapis jednego rejestru (wyniki przedstawiono na rys. 3) oraz odczyt i zapis maksymalnej liczby – 125 – rejestrów (wyniki zaprezentowano na rys. 4).

Ponadto, wygenerowano przypadki mieszane o równej liczbie ramek najdłuższych oraz najkrótszych w dwóch konfiguracjach, różniących się konstrukcją scenariusza wymian. Naprzemienne rozmieszczenie transakcji długich oraz krótkich jest przykładem błędnie przygotowanego scenariusza wymian (na rys. 5 oznaczony jako mieszane – przypadek gorszy).

Zgrupowanie transakcji tej samej długości oznaczono na rys. 5 jako przypadek lepszy. W obu wariantach wskazano wpływ liczby portów szeregowych na czas trwania cyklu sieci przy komunikacji z 10 abonentami.



Rys. 5. Średnie wartości długości cyklu sieci dla różnej liczby portów szeregowych przy 10 abonentach i różnych scenariuszach wymian

Fig. 5. Mean value of the network cycle for different number of serial ports with 10 clients and different transaction schedule

4. Wyniki obserwacji

Analiza wyników symulacji przedstawionych na rys. 3 oraz rys. 4 pokazuje, że rezygnując z topologii magistrali na rzecz koncentratora z jedynie dwoma portami szeregowymi następuje dwukrotne skrócenie czasu trwania cyklu sieci. Jest to wynikiem jednoczesnej komunikacji za pośrednictwem obu portów. Ponieważ czas trwania wszystkich zaplanowanych transakcji jest jednakowy, długość cyklu sieci zbudowanej z koncentratorom ($T_c(i)$) można obliczyć znając czas trwania cyklu sieci w topologii magistrali ($T_c(1)$) według wzoru (2). Przy założo-

nej prędkości transmisji, obciążenie wynikające z przełączania sieci połączeń oraz zarządzania pamięcią mikroprocesora jest znikome.

$$T_C(i) = \frac{T_C(1)}{i}, \quad i = 2, 3, \dots \quad (2)$$

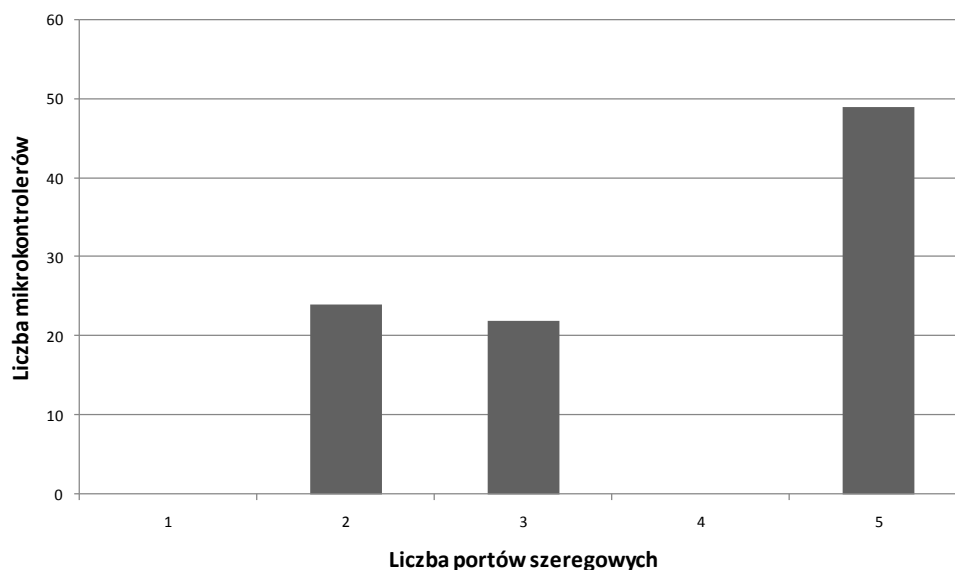
gdzie: i to liczba portów szeregowych, natomiast T_C czas trwania cyklu sieci.

W rzeczywistym systemie węzły udostępniają różne ilości danych, z czego wynikają różnice w długości wymienianych z nimi ramek. Ponieważ koncentrator komunikuje się jednocześnie z kilkoma urządzeniami, według algorytmu, który przedstawiony został w rozdziale 2.3, niezbędne jest oczekiwanie na zakończenie najdłuższej z transakcji. Wyniki przedstawione na rys. 5 pokazują, że zastosowanie koncentratora, gdy scenariusz wymian zawiera naprzemiennie najdłuższe oraz najkrótsze możliwe ramki, generuje czas cyklu sieci o długości równej przypadkowi o samych najdłuższych ramkach. Stosując topologię magistrali, T_C wyraźnie różni się w obu przypadkach.

W przypadku topologii magistrali posortowanie scenariusza wymian względem czasu trwania transakcji nie zmienia wartości T_C . Zachowanie koncentratora jest natomiast trudne do przewidzenia. Przełączanie portów dolnych według zależności (1) powoduje jednoczesne rozpoczynanie transakcji o długich i krótkich ramkach. Podczas analizowanej symulacji do portów dolnych koncentratora podłączono 10 urządzeń. Pięć z nich udostępnia jeden rejestr, natomiast kolejne pięć 125 rejestrów. Użycie pięciu portów szeregowych powoduje jednoczesne przeprowadzanie transakcji tylko o równych czasach trwania. Mniejsza liczba portów powoduje czasem jednoczesne wystąpienie transakcji o różnych czasach trwania, większa natomiast gwarantuje, że jednocześnie wystąpią oba rodzaje transakcji.

Powyżej pięciu portów szeregowych, występujących w koncentratorze, w systemie z 10 abonentami, zysk wynikający ze skrócenia długości czasu cyklu sieci jest znikomy. Należy ponadto zauważyć, że producenci układów scalonych dostarczają mikrokontrolery o określonej liczbie kontrolerów UART. W rodzinie STM32F1 jest ona ograniczona do trzech konfiguracji o liczebności przedstawionej na rys. 6.

W przypadku konieczności zastosowania większej liczby kontrolerów UART rozwiązaniem jest stosowanie układów FPGA w miejsce sieci połączeń oraz implementacja w tej logice programowalnej bloków opisu sprzętu realizujących ich zadania. Opublikowany przez społeczność *OpenCores.org* blok *Minimal UART Core* zajmuje 64 elementy slice struktury Xilinx Spartan 3E, co w przypadku układu XC3S400 pozwala na uruchomienie 56 kontrolerów. Zastosowanie układu FPGA eliminuje konieczność wykorzystania osobnego procesora, możliwa jest implementacja rdzenia bezpośrednio w logice programowalnej [4].



Rys. 6. Rozkład liczby portów szeregowych w rodzinie mikrokontrolerów STM32F1

Fig. 6. Distribution of serial port's number in microcontroller STM32F1 family

W artykule [5] przedstawiono inne podejście do elastycznego projektowania topologii sieci z użyciem gotowych urządzeń. Wykorzystanie łączności bezprzewodowej pozwoliło w prosty sposób połączyć węzły w topologii hierarchicznej. Użyte w tym celu bramy komunikują się z wieloma abonentami, jednak mają wyłącznie jeden interfejs bezprzewodowy oraz są przezroczyste dla protokołu Modbus. Nie magazynują zatem danych, co nadaje proponowanemu systemowi cechy magistrali.

5. Podsumowanie

Przeprowadzone badania pokazały, że zastosowanie koncentratora z nawet minimalną liczbą portów szeregowych skraca wyraźnie czas trwania cyklu sieci w stosunku do topologii magistrali. Podejście takie wymusza jednak na projektancie odpowiednie skonstruowanie scenariusza wymian i dopasowanie go do architektury wewnętrznej koncentratora. Rezygnując z systemu mikroprocesorowego na rzecz logiki programowalnej, konstruktor może znacznie zwiększyć liczbę portów szeregowych, jednak jak pokazały symulacje nie zawsze przynosi to znaczący zysk. Korzystając z symulacji komputerowej możliwe jest sprawdzenie działania sieci już na etapie jej projektowania, a środowisko Omnet++ pozwala na emulację zachowania dowolnych, pracujących w niej urządzeń.

BIBLIOGRAFIA

1. Modbus application protocol specification v1.1b
2. Kwiecień A.: Analiza przepływu informacji w komputerowych sieciach przemysłowych. Pracownia Komputerowa Jacka Skalmierskiego. Gliwice 2000.
3. Kluk P.: Wykorzystanie protokołu komunikacyjnego MOD BUS RTU w urządzeniach EA Z bazujących na platformie ARM, [w:] Elektronika, nr 5, 2007.
4. Cupek R., Piękoś P., Poczobutt M., Ziębiński A.: FPGA Based “Intelligent Tap” Device for Real-Time Ethernet Network Monitoring [in:] Kwiecień, A., Gaj, P., Stera, P. (eds.): CN2010, CCIS 79, Springer, Heidelberg (2010), pp. 58-66.
5. Guarese G., Sieben F., Webber T., Dillenburg M., Marcon C.: Exploiting Modbus Protocol in Wired and Wireless Multilevel Communication Architecture, [in:] Computing System Engineering (SBESC), 2012 Brazilian Symp. on, 5-7 Nov. 2012, pp. 13-18.

Wpłynęło do Redakcji 20 marca 2013 r.

Abstract

This article shows the influence of hub hardware configuration on network cycle time. Paragraph 2 describe position of the hub in control system and its internal structure. Moreover, equation (1) presents the method of switching signals between serial and lower ports.

As the result of simulations of the network model, in the fig. 3 and 4 show dependency between network cycle time, numbers of nodes and amount of UARTs. Simulations were repeated with different transaction schedule.

The analysis pointed to the network with hub as always having shorter network cycle time than the bus topology, but there is a limit of profitability. Huge number of serial ports in microcontroller is rare property but FPGA modules could be used instead.

Adres

Łukasz HERB: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,
44-100 Gliwice, Polska, lukasz.herb@polsl.pl