

Lucjan PELC

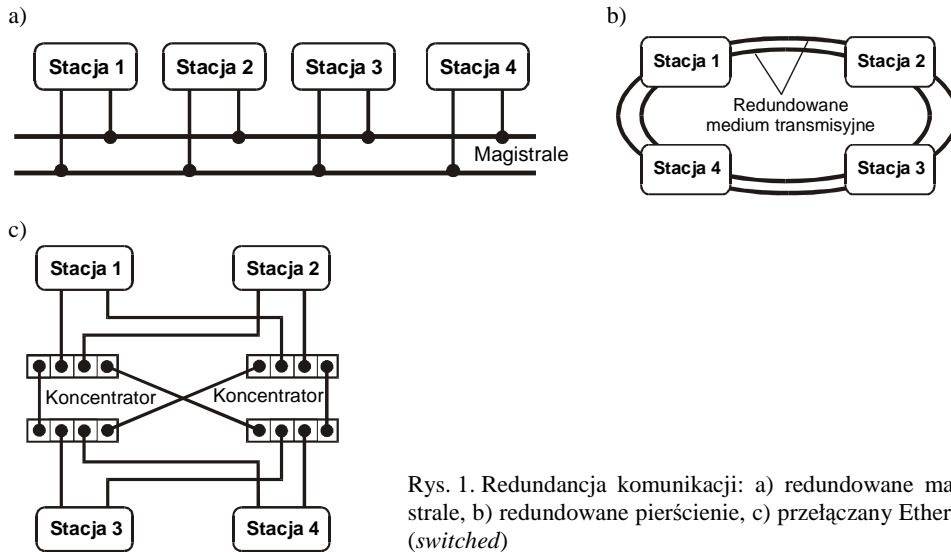
Politechnika Rzeszowska

KOMUNIKACJA W NIEWIELKICH SYSTEMACH KONTROLNO-POMIAROWYCH

W artykule przedstawiono zagadnienie wymiany danych w ramach niewielkiej sieci urządzeń aparaturowych, realizujących zadania związane z cyklicznym pomiarem, sterowaniem czy regulacją. Komunikacja w takim systemie musi uwzględniać ograniczenia czasowe na terminowość wymiany danych. Ponadto, ze względu na zadania takiego systemu, komunikacja bywa cykliczna lub wyzwalana. W związku z tym zaproponowano protokół komunikacyjny, który łączy zalety obu tych rodzajów komunikacji, gwarantując jednocześnie zachowanie stałego cyklu komunikacyjnego. Warstwa fizyczna przedstawionego protokołu może być oparta na ciągle bardzo popularnym w zastosowaniach przemysłowych standardzie RS-485 albo CAN. W artykule wskazano także globalne parametry komunikacyjne, które pozwalają określić ograniczenia czasowe niesione przez dany protokół. Zaprezentowano praktyczne obliczenia dla protokołów typu *master-slave*, wchodzących w skład proponowanego protokołu z przepływem.

1. Wprowadzenie

Systemy sterowania instalowane w przemyśle są z reguły systemami rozproszonymi. Standardowe protokoły komunikacyjne dla magistral polowych (*fieldbus*) stosowanych w systemach znanych firm, takie jak Profibus DP, ControlNet, Foundation Fieldbus (FF) czy Factory Information Protocol (FIP) [1-3], zapewniają wymagany poziom determinizmu czasowego i otwartość na urządzenia innych producentów. Jednocześnie w większych zastosowaniach, np. w energetyce, wobec systemów sterowania stawia się znacznie wyższe wymagania dotyczące dyspozycyjności (*availability*). Sprowadzają się one przede wszystkim do redundancji (podwojenia) jednostek centralnych i podsystemu komunikacyjnego [4-6]. W przemyśle dominują jednak systemy małe i średnie (do 1000 sygnałów). Ograniczone możliwości sprzętowe małych systemów zmuszają do opracowania własnych, specjalizowanych protokołów komunikacyjnych, zapewniających wymaganą efektywność, mimo mniej zaawansowanego sprzętu. Otwartość systemu posługującego się własnym protokołem polowym powinny umożliwiać odpowiednie konwertery komunikacyjne.



Rys. 1. Redundancja komunikacji: a) redundowane magistrale, b) redundowane pierścienie, c) przełączany Ethernet (*switched*)

Coraz większe wymagania odnośnie dyspozycyjności nakładają również na małe systemy potrzebę przynajmniej ograniczonej redundancji. Na rysunku 1. pokazano typowe rozwiązania redundancji w obszarze zwielokrotnienia medium transmisyjnego.

2. Globalne parametry komunikacji

Chcąc dokonać oceny właściwości czasowych komunikacji prowadzonej według danego protokołu, należy wcześniej zdefiniować odpowiednie parametry. Dalej wskazano kilka z nich, które posłużą do oceny porównawczej dwu protokołów.

Czas transakcji. Określa się go odmiennie dla różnych mechanizmów dostępu do medium. W uproszczeniu można jednak powiedzieć, że jest to czas potrzebny na zrealizowanie sekwencji *wysłanie polecenia – odebranie odpowiedzi*. Ponieważ parametr ten ma podstawowe znaczenie podczas określania charakterystyki czasowej systemu z protokołem magistrali polowej, metoda jego wyznaczania zostanie dokładniej omówiona. Przykładowo rozważono komunikację *Master-Slave*. Transakcja przebiega wówczas w następujących krokach, z czasami podanymi po pauzie:

- stacja *Master* przygotowuje polecenie (*query*) – T_{PFQ} ,
- ramka polecenia jest transmitowana za pośrednictwem magistrali – T_{TFQ} ,
- stacja *Slave* odbiera i analizuje polecenie – T_{AFQ} ,

- *Slave* przygotowuje odpowiedź (*response*) – T_{PFR} ,
- transmisja odpowiedzi – T_{TFR} ,
- stacja *Master* odbiera i analizuje odpowiedź – T_{AFR} .

Suma czasów z tych kroków daje czas trwania transakcji:

$$T_T = T_{PFQ} + T_{TFQ} + T_{AFQ} + T_{PFR} + T_{TFR} + T_{AFR} \quad (1)$$

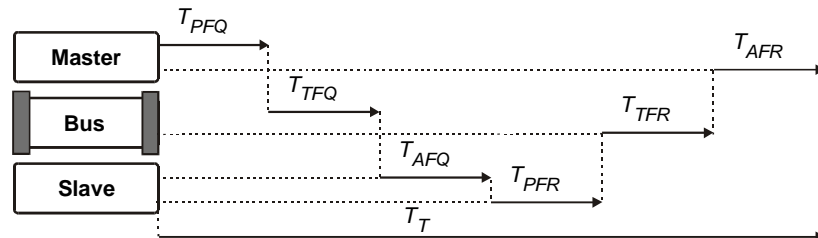
Graficznie przedstawia go rys. 2.

Należy dodać, że *czas analizy ramki* zapytania T_{AFQ} i odpowiedzi T_{AFR} obejmuje *detekcję ramki*, jej *rozpoznanie* (analizę) i dodatkowo, w zależności od konkretnej realizacji, *czas cyklu programu automatu komunikacyjnego*. Przyjęto, że *czas analizy ramki* jest to czas, jaki upływa od chwili jej odebrania do momentu przygotowania danych dla wyższej warstwy oprogramowania.

Czas cyklu. Podobnie jak dla czasu transakcji, określenie *czasu cyklu* zależy od przyjętego mechanizmu dostępu. Ogólnie *czas cyklu* należy rozumieć jako czas, który upływa do uzyskania przez daną stację kolejnego dostępu do medium. Jeśli założyć, że w systemie w każdym cyklu komunikacyjnym realizowane jest N transakcji, wówczas *czas cyklu* T_C wyraża wzór:

$$T_C = \sum_{i=1}^N T_{Ti} \quad (2)$$

gdzie T_{Ti} jest czasem trwania i -tej transakcji.



Rys. 2. Czas trwania pojedynczej transakcji w komunikacji *Master-Slave*

Sprawność sieci. Definiuje się ją odmiennie dla różnych sieci oraz celu, któremu mają służyć. Przykładowo w sieciach z protokołem FIP sprawność określa się jako stosunek całkowitego czasu trwania transmisji danych¹ do cał-

¹ Chodzi tu o dane użytkowe, np. stan wejść i wyjść obiektowych danej stacji procesowej.

kwitego czasu transakcji wymiany danych. Jeśli przyjąć, że spośród wszystkich transakcji tylko K dostarcza danych, to sprawność będzie wyrażana wzorem:

$$\eta_{(1)} = \frac{\sum_{i=1}^K T_{TData,i}}{\sum_{i=1}^K T_{Ti}} 100\% \quad (3)$$

gdzie $T_{TData,i}$ – czas transmisji danych w j -tej transakcji.

Tak określona sprawność informuje, jak szybko jest realizowany cykl wymiany ogółu danych. W przypadku gdy transmisja danych odbywa się w każdej transakcji, całkowity czas transakcji jest równy czasowi cyklu sieci:

$$\sum_{j=1}^N T_{TData,j} = T_C.$$

Prostszym określeniem sprawności jest wzór [7]:

$$\eta_{(2)} = \frac{T_{TData}}{T_T} 100\% \quad (4)$$

gdzie T_T ma postać z równania (1).

Bardzo często można spotkać sprawność określoną jako:

$$\eta_{(3)} = \frac{T_{TData}}{T_{TFData}} 100\% \quad (5)$$

gdzie T_{TFData} oznacza czas transmisji ramki z danymi w pojedynczej transakcji.

Ponieważ $T_{TFData} = T_{TFQ} + T_{TFR}$, więc $T_T > T_{TFData}$ (zob. wzór (1)), $\eta_{(3)} > \eta_{(2)}$. Producenci najchętniej podają sprawność w formie $\eta_{(3)}$. Taka definicja czasami może wprowadzać w błąd, informuje bowiem jedynie o narzucie czasowym warstwy łącza.

Przepustowość sieci. Definicja jest powiązana ze sprawnością. Przykładowo dla $\eta_{(2)}$ (jak w równaniu (4)) przepustowość określa się wzorem:

$$P_{(2)} = \frac{L_{Data} L_U}{T_{Ti}} \text{ [kb/s]} \quad (6)$$

gdzie: L_{Data} – liczba znaków danych, L_U – liczba bitów znaku, T_T – czas transakcji.

Dla $\eta_{(3)}$ z równania (5) przepustowość ma postać:

$$P_{(3)} = \frac{L_{Data} L_U}{T_{TF_Data}} \text{ [kb/s]} \quad (7)$$

gdzie: L_{Data} – liczba znaków danych, L_U – liczba bitów znaku, T_{TFData} – czas transmisji ramki z danymi.

Podobnie jak w przypadku sprawności, również dla przepustowości zachodzi relacja $P_{(3)} > P_{(2)}$.

W celu poprawy parametrów czasowych komunikacji stosuje się *grupowanie* i *szeregowanie* komunikatów, a w systemach rozległych dodatkowo podział na podsieci. Grupowanie kilku komunikatów w jeden zmniejsza obciążenie systemu, a odpowiednie szeregowanie pozwala dochować reżimy czasowe podczas przeciążenia. W systemach małych i średnich wystarcza na ogół grupowanie komunikatów przez stację, z której pochodzą dane. Szeregowanie może być naturalne, tzn. zgodne z kolejnością pojawiania się pytań, albo priorytetowe.

3. Obliczenia dla protokołów Modbus i Trans

Jednym z najbardziej popularnych protokołów jest protokół Modbus w wersji RTU. Można go spotkać w większości urządzeń aparaturowych. W proponowanym protokole TMI komunikacja wyzwolana (*Master-Slave*) może odbywać się według zmodyfikowanego Modbusa RTU lub protokołu Trans [8, 9]. Zasadne jest zatem porównanie obu tych protokołów. Dalej rozważono przypadek cyklicznego odczytywania z urządzenia kontrolno-pomiarowego 6 wejść i 2 wyjść analogowych oraz 16 wejść i 16 wyjść binarnych.

Modbus RTU. Wprowadzona modyfikacja polega na tym, że wejścia i wyjścia analogowe wyrażane jako 4-bajtowe liczby zmiennoprzecinkowe (*float*) są reprezentowane przez 2-bajtowe rejestry. Ponadto grupowanie dostępne jest tylko dla danych tego samego typu (bitów lub rejestrów), a to oznacza, że potrzebne są 4 transakcje, aby odczytać wszystkie wejścia i wyjścia urządzenia. Odczyt wejść i wyjść binarnych (bitów) umożliwiają odpowiednio funkcje Modbusa 01 i 02 hex, wejść i wyjść analogowych (w artykule „rejestrów”) – funkcje 03, 04 hex. Ze względu na popularność Modbusa dodatkowe informacje dotyczące ramki zostaną pominięte. Znaczenia kolejnych bajtów ramki dla wymienionych funkcji podano w tab. 1.

Tabela 1. Zawartość ramki dla funkcji 01, 02, 03 i 04 hex w protokole Modbus

Funkcje: 01, 02 hex				03, 04 hex			
Pytanie		odpowieź		pytanie		odpowieź	
Bajt	opis	bajt	opis	bajt	opis	bajt	opis
1	adres sterownika	1	adres sterownika	1	adres sterownika	1	adres sterownika
2	nr funkcji	2	nr funkcji	2	nr funkcji	2	nr funkcji
3, 4	adres bitu	3	liczba bajtów	3, 4	adres rejestru	3	liczba bajtów
5, 6	liczba bitów	4 do n	zgrupowane bity	5, 6	liczba rejestrów	4 do n	2-bajtowe rejestry
7, 8	suma CRC	$n+1$, $n+2$	suma CRC	7, 8	suma CRC	$n+1$, $n+2$	suma CRC

Trans. Ramkę protokołu przedstawia rys. 3., gdzie kolejne pola oznaczają: SOH – znak początku komunikatu (7E hex), CNO – kod komunikatu i numer sterownika (starsza i młodsza część bajtu), DATA – bajty danych, CRC suma kontrolna.

SOH	Adres	CNO	DATA ...	CRC
8 b	8 b	8 b	$n \times 8$ b	16 b

Rys. 3. Ramka niestandardowego protokołu Trans (b≡bit)

Możliwy jest grupowy odczyt danych różnego typu. Pozwala na to funkcja (usługa) DSP (60 hex) w trybie IOV (30 hex). Wymiana danych rozważanych w przykładzie odbywa się więc w jednej transakcji, co zilustrowano w tab. 2.

Tabela 2. Zawartość ramki dla funkcji DSP, tryb IOV – protokół Trans

Funkcja: DSP (60 hex) dla trybu IOV (30 hex)					
Pytanie		odpowieź			
Bajt	opis	bajt	opis	bajt	opis
1	SOH	1	SOH	35, 36	16 wejść binarnych
2	DSP + nr	2	DSP + nr	37, 38	16 wyjść binarnych
3	rodzaj trybu – IOV	3 do 26	8 wejść analogowych	39, 40	suma kontrolna
4, 5	suma CRC	27 do 34	2 wyjścia analogowe	Dane analogowe są 4-bajtowymi liczbami <i>float</i> .	

W celu porównania obu protokołów założono, że komunikacja odbywa się w systemie złożonym z komputera nadrzędnego i $N = 9$ stacji procesowych.

Ponadto każdy znak ramki jest kodowany na $L_U = 11$ bitów (typowa wartość), a prędkość komunikacji wynosi $V = 4800$ b/s. W tabeli 3. zebrano wyniki obliczeń czasu transakcji T_T , cyklu T_C , sprawności $\eta_{(2)}$ i $\eta_{(3)}$ oraz przepustowości $P_{(2)}$ i $P_{(3)}$. W obliczeniach wykorzystano wzory (1)-(7). W celu jednolitości założono dla obu protokołów $T_{PFQ} + T_{AFR} = 5$ ms (czas przygotowania zapytania przez komputer + czas analizy odebranej odpowiedzi) i $T_{AFQ} + T_{PFR} = 10$ ms (czas analizy zapytania przez stację + czas przygotowania odpowiedzi).

Podane sprawności i przepustowości wykazują przewagę niestandardowego protokołu Trans nad Modbusem. Trans pozwolił na prawie dwukrotne skrócenie cyklu komunikacji, mimo większej precyzji transmitowanych danych analogowych (*float* zamiast *integer*). Sprawność i przepustowość zwiększyła się średnio półtora raza.

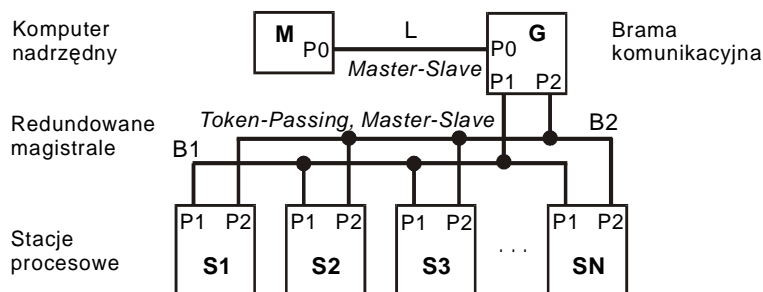
Tabela 3. Wyniki obliczeń dla protokołów Modbus i Trans

Protokół	Funkcja [hex]	L_{Data} [bajt]	$L_Q + L_R$ [bajt]	T_T [ms]	T_C [s]	$\eta_{(2)}$ [%]	$P_{(2)}$ [kb/s]	$\eta_{(3)}$ [%]	$P_{(3)}$ [kb/s]	
Modbus	01 i 02	$4 + 3 = 7$	$8 + 7 = 15$	49.4	221	1.99	33	1.6	47	2.2
	03	$4 + 13 = 17$	$8 + 17 = 25$	72.3			54	2.6	68	3.3
	04	$4 + 5 = 9$	$8 + 9 = 17$	54			38	1.8	53	2.5
Trans	60	$1 + 36 = 37$	45	118	1.06	72	3.5	82	4	

4. Protokół z przeplotem mechanizmów dostępu do magistrali

Opracowany przez autora protokół TMI (*Token-Passing and Master-Slave Interleaving Protocol*) jest przeznaczony dla niewielkiego systemu sterowania, gdzie dyspozycyjność komunikacji ma istotne znaczenie [10]. Cecha ta jest niezbędna podczas obsługi tych procesów technologicznych, w których usterka systemu komunikacyjnego może prowadzić do zatrzymania pracy całej instalacji. W protokole TMI dyspozycyjność osiągnięto przez wprowadzenie redundancji magistrali z automatycznie przełączanym trybem pracy, co pozwala na realizację *stopniowanej degradacji*. TMI wykorzystuje redundowaną magistralę połową (*fieldbus*) oraz *przeplatany dostęp* do medium (*interleaving*), łączący mechanizmy *Token-Passing* i *Master-Slave*. Rysunek 4. przedstawia system rozproszony komunikujący się według TMI, którego elementami składowymi są:

- komputer nadrzędny M (*Master*) z portem P0 (*Port*) i linią L (*Line*),
- brama G (*Gate*) z redundowanymi magistralami B1, B2 (*Bus*) oraz portami P0, 1 i 2,
- stacje procesowe S (*Station*), każda z dwoma portami P1, P2.



Rys. 4. System z protokołem TMI

W zależności od dostępnych w danej chwili zasobów sprzętowych komunikacja w TMI odbywa się w jednym z czterech podstawowych trybów:

- praca normalna – wszystkie elementy systemu są sprawne,
- odłączenie jednej magistrali – druga pełni rolę magistrali wiodącej,
- odłączenie bramy komunikacyjnej – trwa jedynie komunikacja *Token-Passing*,
- odłączenie bramy i magistrali – *Token-Passing* za pośrednictwem magistrali wiodącej.

W artykule zostanie omówiony jedynie tryb pracy normalnej sieci komunikującej się wg TMI [11].

Podstawowe cechy. Zakłada się, że w systemie są sprawne wszystkie elementy, a rolę magistrali wiodącej pełni B1, rezerwowej zaś B2 (por. rys. 4.). Za pośrednictwem magistrali B1 jest prowadzona zarówno komunikacja między stacjami procesowymi S1,...SN, jak i komunikacja między nimi a komputerem nadrzędnym M (przeplatanie). Magistrala rezerwowa służy do przesyłania komunikatów testowych. Komunikaty w obu magistralach są przesyłane synchronicznie. Brama G pośredniczy między komputerem i stacjami, przez co dla komputera M cały system przyjmuje postać typu *Master-Slave*. W ramach linii L łączącej komputer z bramą komunikacja odbywa się według protokołu Modbus RTU. Komunikacja komputer – stacja kontrolno-pomiarowa będzie nazywana *pionową (vertical)*, a między równorzędnymi stacjami – *poziomą (horizontal)* [8].

Transakcja normalna. Na rysunku 5. pokazano wymianę komunikatów dla linii L i magistral B1, B2, wprowadzając następujące oznaczenia:

1) komunikaty

- $q(i)$ – zapytanie *Mastera* skierowane do stacji i (*query*),
- $r(i)$ – odpowiedź ze stacji i do *Mastera* (*response*),
- $m(i,j)$ – dane oraz znacznik ze stacji i do j (*message*),
- $t(i,j)$ – komunikat testowy ze znacznikiem ze stacji i do j (*test frame*),

2) czasy

- T_{TH} – transakcja komunikacji poziomej,
- T_{TV} – transakcja komunikacji pionowej,
- T_{SV} – odstęp pomiędzy $m(i,j)$ a $q(i)$ lub $r(i)$ (cisza – *silence*).

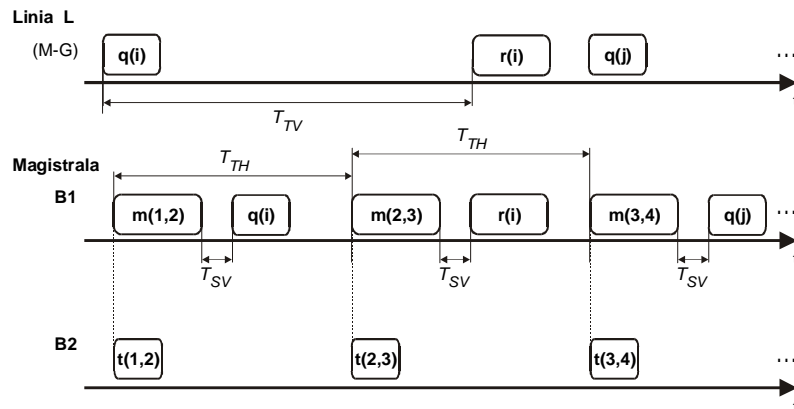
Zakłada się, że czas transakcji komunikacji poziomej T_{TH} jest stały. Za-chodzi zatem związek:

$$\max T_{Tm(i,j)} + 2T_{SV} + \max T_{T\{q(i),r(i)\}} \leq T_{TH} \quad (8)$$

gdzie: $\max T_{Tm(i,j)}$ – maksymalny czas transmisji ramki komunikatu poziomego,

$\max T_{T\{q(i),r(i)\}}$ – maksymalny czas transmisji komunikatu pionowego.

W protokole TMI zrezygnowano z odrębnych komunikatów wysyłania znacznika i potwierdzania odbioru. Przyjęto, że komunikat z danymi zawiera również znacznik, a potwierdzeniem jest wysłanie własnego komunikatu przez stację wskazaną znacznikiem. Zmniejsza to ruch generowany na magistrali.



Rys. 5. Normalna wymiana komunikatów za pośrednictwem linii L i magistral B1, B2

Na podstawie analizy transakcji komputer nadrzędny M – stacja procesowa Si (rys. 5.) można wyróżnić jej cztery etapy:

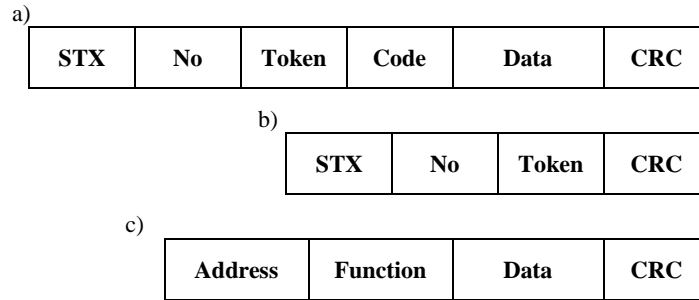
1. Do bramy G za pośrednictwem linii L trafia zapytanie $q(i)$ skierowane z komputera M do stacji Si. Brama nie może od razu przekazać go na magistralę wiodącą B1, obowiązuje bowiem zasada przeplotu *Token-Passing* i *Master-Slave*.
2. Odczekawszy, aż na magistrali B1 zakończy się komunikat poziomy, brama G po dodatkowym czasie ciszy T_{SV} przekazuje magistralą zapytanie $q(i)$ do stacji Si. Komunikat $m(1,2)$ zawiera znacznik przeznaczony dla stacji S2.
3. Stacja Si wstrzymuje się z udzieleniem odpowiedzi, oczekując na pojawienie się na magistrali B1 komunikatu poziomego (przeplot). Ponieważ znacznik jest obecnie w posiadaniu stacji S2, po upływie czasu T_{TH} od momentu pojawienia się komunikatu $m(1,2)$ wysyła ona własny komunikat $m(2,3)$, przekazując znacznik do S3. Równocześnie S2 przesyła magistralą rezerwową B2 komunikat testowy $t(2,3)$. Dopiero po nim oraz po upływie czasu T_{SV} stacja Si przystępuje do wysłania odpowiedzi $r(i)$ na zapytanie $q(i)$.
4. Odebrawszy komunikat $r(i)$ z odpowiedzią stacji Si brama G niezwłocznie przekazuje go do komputera. Jest to zakończenie transakcji.

Jak widać, komunikaty pionowe (*Master-Slave*) pojawiają się na magistrali wiodącej pomiędzy komunikatami poziomymi (*Token-Passing*), realizując przeplatanie mechanizmów dostępu do medium. Ponieważ czas T_{TH} jest dłuższy od czasów transmisji komunikatów poziomych i pionowych ($m(i,j)$, $q(i)$, $r(i)$), czas transakcji komunikacji poziomej jest stały. Dodatkowo ze wzoru (2) opisującego czas cyklu jako sumę czasów transakcji wynika, że dla tej komunikacji również czas cyklu T_{CH} nie ulega zmianie.

5. Ramki komunikatów TMI dla standardów RS-485 i CAN

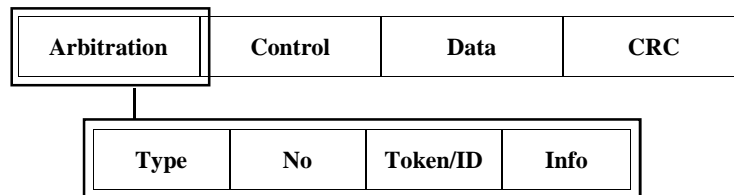
RS-485. Jest to standard komunikacji szeregowej od lat stosowany w przemysłowych protokołach komunikacyjnych (Modbus, Profibus) [2, 5]. Na rysunku 6a przedstawiono zawartość ramki protokołu TMI dla komunikatów poziomych, a na rys. 6b dla testowych. Przyjęto następujące oznaczenia: STX – znak początku, No – numer stacji procesowej – nadawca, Token – znacznik – adres wskazanej stacji, Code – kod interpretacji danych, Data – przesyłane dane, CRC – suma kontrolna.

Komunikacja pionowa wykorzystuje protokół Modbus RTU lub protokół Trans (por. pkt 2.). Uproszczoną ramkę komunikatów pokazano na rys. 6c, pomijając czas ciszy na magistrali pełniący rolę znacznika początku i końca. Address jest adresem odpytywanej stacji (*Slave*), a Function określa funkcję protokołu (usługę). Przykłady omówiono w tab. 1. Data zawiera przekazywane dane, CRC jest zaś sumą kontrolną.



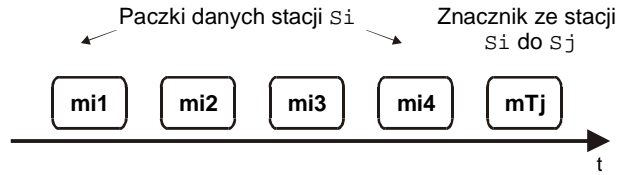
Rys. 6. Ramki komunikatów protokołu TMI: a) poziomy $m(i,j)$, b) testowy $t(i,j)$, c) pionowy $q(i)$, $r(i)$

CAN. Standard ten, oprócz fizycznych parametrów komunikacji, określa także mechanizm dostępu do medium [12]. Protokół TMI wprowadza determinizm co do kolejności nadawania komunikatów przez stacje, uzupełniając w ten sposób CAN-owskie *porównanie binarne*. Na rysunku 7. podano ramkę CAN-a, rozwijając obszar arbitrażu (*arbitration*), najistotniejszy z punktu widzenia wprowadzanego przez protokół TMI mechanizmu dostępu. Nowe oznaczenia są następujące: Arbitration – arbitraż, Control – obszar kontrolny, Type – typ komunikatu, Token/ID – znacznik/adres stacji dla Modbusa, Info – uzupełnienie typu komunikatu.



Rys. 7. Wspólna ramka komunikatów protokołu TMI – poziomy $m(i,j)$, testowy $t(i,j)$ i pionowy $q(i)$, $r(i)$ – dla standardu CAN z rozwiniętym obszarem arbitrażu

Ze względu na ograniczenie rozmiaru pola danych CAN-a do 8 bajtów protokół TMI wyjątkowo dopuszcza odrębne komunikaty dla znacznika i danych. W jednej transakcji może być przekazane do 4 ramek z danymi, jako tzw. paczki danych. Stacja po otrzymaniu znacznika wysyła własne dane pogrupowane w paczki, a po nich komunikat ze znacznikiem wskazującym na kolejną stację w systemie. Dotychczasowy komunikat $m(i,j)$ składa się więc teraz z ramek danych $mi1$ do $mi4$ oraz ramki mTj ze znacznikiem dla stacji następnej, co obrazuje rys. 8. Komunikatom testowym $t(i,j)$ i pionowym $q(i)$, $r(i)$ nadal odpowiadają pojedyncze ramki.



Rys. 8. Komunikat poziomy $m(i,j)$ do realizacji w standardzie CAN

Symboliczny sposób kodowania komunikatów TMI w obszarze arbitrażu CAN-a podano w tab. 4., przy czym pominięto pole No, w którym jest zawsze podawany adres nadawcy. Pola arbitrażu z rys. 7. dla kolejnych ramek komunikatu $m(i,j)$ z rys. 8. będą następujące:

mi1. Pole Type informuje, że w ramce są przesyłane dane, Token/ID zawiera numer paczki (tutaj 1), No określa adres nadawcy i , Info mówi o rodzaju danych (binarne, analogowe – *float*).

mi1, mi2, mi3, mi4. Znaczenie pól ramek jest analogicznie jak dla $mi1$, z tym że Token/ID zawiera odpowiednio numery 2, 3, 4 (numery ramki w paczce danych).

mTj. Jest to komunikat ze znacznikiem, o czym informuje zawartość pola Type = znacznik. Wartość znacznika jest umieszczona w polu Token/ID = j . Ponieważ ten rodzaj komunikatu nie przekazuje danych, Info = brak danych.

Tabela 4. Zawartość obszaru arbitrażu dla komunikatów poziomych, pionowych i testowych

Komunikacja	Pole arbitrażu (<i>arbitration</i>)		
	Type	Token/ID	Info
Pozioma	dane	nr paczki	rodzaj danych
	znacznik	znacznik	brak danych
Pionowa	zapytanie	adres stacji	–
	odpowiedź	adres bramy	–
Test	test	adres testowanej stacji	test poziomy

O wyborze standardu CAN jako drugiej realizacji protokołu TMI zdecydowała jego relatywnie duża prędkość transmisji. W typowych zastosowaniach wynosi ona 500 kb/s, maksymalną zaś prędkością w standardzie RS-585 jest 19,2 kb/s. Ponad 25-krotne przyspieszenie ułatwia dotrzymanie czasu transakcji T_{TH} zarówno w trakcie przepłotu, jak i podczas przełączania magistral (por. rys. 5.).

6. Podsumowanie

W artykule rozpatrzono problem specyfikacji niestandardowego protokołu komunikacyjnego przeznaczonego dla niewielkich systemów kontrolno-pomiarowych, gdzie zasadniczą rolę odgrywa terminowość komunikacji oraz stabilny cykl. Dla ilościowego określenia właściwości czasowych danego protokołu wskazano globalne parametry komunikacyjne, a następnie wykorzystano je praktycznie do porównania i oceny dwu protokołów klasy *Master-Slave*, które wchodzi w skład proponowanego rozwiązania. Proponowany protokół TMI łączy w ramach przepływu dwa mechanizmy dostępu do medium transmisyjnego: *Token-Passing* i *Master-Slave*. Omówiono implementację TMI w ramach standardu RS-485 oraz CAN, podając odpowiednie ramki komunikatów.

Literatura

- [1] FIP Toolbox. General Introduction. Reference Manual. WorldFIP, Nancy 1995.
- [2] EXPERION PKS CEE-based Controller Specifications and Models, Honeywell, January 2003.
- [3] Kwiecień A.: Analiza przepływu informacji w komputerowych sieciach przemysłowych. Wydawnictwo Pracowni Komputerowej, Gliwice 2001.
- [4] Lawrenz W.: CAN System Engineering. Springer, Berlin 1997.
- [5] Pelc L.: Drivery komunikacyjne FIX-PSW/WWT. Protokoły Trans i Modbus. Raport z prac badawczo-wdrożeniowych. KAI PRz, Rzeszów 1996.
- [6] Pelc L., Trybus L.: Specyfikacja formalna prostych protokołów magistral polowych. Pomiary Automatyka Kontrola, no 1, 2005, 14-17.
- [7] Pimentel J.R.: Communication networks for manufacturing. Prentice-Hall Int., 1990.
- [8] Sacha K.: Sieci miejscowe PROFIBUS. Wydawnictwo MIKOM, Warszawa 1998.
- [9] SIEMENS: TELEPERM XP. The process control system for economic power plant control. Catalog LT 600, May 2002.
- [10] Storey N.: Safety-Critical Computer Systems. Addison-Wesley, New York 1996.
- [11] Teleperm XP. The process control system for economic power plant control. Cat. LT 600, Siemens, 2002.
- [12] Trybus L.: Regulatory wielofunkcyjne. WNT, Warszawa 1998.

COMMUNICATION IN SMALL-SCALE CONTROL AND MEASUREMENT SYSTEMS

Summary

This paper presents the problem of data exchange within a small network. Each node in the network performs the task of cyclic measurement and control. Communication in the system must take into account the time constraints on timeliness of data exchange. Furthermore, due to the task of such a system, communication is sometimes cyclic or triggered. Therefore, the proposed communication protocol, which combines the advantages of both of these types of communications,

while guaranteeing to maintain constant communication cycle. The paper also indicated global communication parameters that allow to set time limits carried by the protocol. Practical calculations are presented for master-slave protocols, which are used in the proposed protocol.

DOI: 10.7862/re.2012.2