

Roman MIELCAREK
Politechnika Poznańska, Instytut Informatyki

MODUŁ PROGRAMOWY SIECI PRZEMYSŁOWEJ K:K

Streszczenie. Artykuł przedstawia dotychczasowy stan prac nad programowym modułem sieci przemysłowej o nazwie K:K. Koncepcja tej sieci powstała na bazie analizy działania sieci wbudowanych w oprogramowanie systemowe kompaktowych sterowników FX firmy Mitsubishi Electric. Charakteryzuje się ona tym, że nie ma ona stacji Master, a kolejność transmisji danych w sieci oparta jest na tokenie czasowo-adresowym.

Słowa kluczowe: sterownik PLC, sieć przemysłowa

INDUSTRIAL NETWORK SOFTWARE MODULE K:K

Summary. This paper presents the current status of the software module of the industrial network called the K: K. The concept of this network was based on analysis of the performance of the networks which are embedded in the compact PLC FX system software from Mitsubishi Electric. It is characterized by the fact that it has no master station and the sequence of data transmission is based on the time-address token.

Keywords: PLC, industrial network

1. Możliwości komunikacyjne sterowników rodziny FX

Rodzina kompaktowych sterowników o symbolu FX, rozwijana od początku lat 90. przez firmę Mitsubishi Electric, ma w swoim zestawie dwa najnowsze produkty. Jednym z nich jest najszybszy sterownik o symbolu FX3U [3, 4, 7], o czasie wykonania instrukcji podstawowej 0,065 μ s, należący do tzw. rodziny głównej. Drugim produktem jest najnowszy sterownik, oznaczony symbolem FX3G [6, 7], którego zadaniem jest zastąpienie dwóch istniejących dotychczas podrodzin o symbolach FX1S i FX1N.

Wszystkie sterowniki rodziny FX mają po swojej lewej stronie wbudowany interfejs komunikacyjny, który z dotychczasowego, pojedynczego zasadniczo interfejsu szeregowego (FX1S, FX1N i FX2N) został przekształcony w sterownikach FX3U i FX3G w uniwersalny interfejs wielomodułowy, do obsługi modułów analogowych włącznie. Interfejs ten pozwala podłączyć do dwóch modułów szeregowych na ogólną liczbę do 6 modułów w FX3U. Moduły szeregowy mogą być wykonane w standardzie RS232/422/485 i mogą tworzyć podstawowy (bez numeru) i dodatkowy (nr 2) porty szeregowy. Podstawowy port szeregowy może zostać wykorzystany do komunikacji szeregowy w tzw. sieciach wbudowanych, zaimplementowanych w systemie operacyjnym sterownika. Zarówno port podstawowy, jak i dodatkowy mogą wymieniać dane według dedykowanych (wbudowanych) protokołów komunikacyjnych sterownika FX, a także według dowolnie realizowanego protokołu komunikacyjnego. Sieci wbudowane pracują na modułach w standardzie RS485 [5]. Do nich należą:

- połączenie równoległe „1:1” (ang. parallel link) dwóch sterowników,
- sieć „N:N”, umożliwiająca połączenie do 8 sterowników.
- programowa sieć typu wielopunkt (ang. multidrop), bazująca na protokole dedykowanym do komunikacji, np. z przetwornicami częstotliwości (falownikami).

Na rys. 1 przedstawiono zdjęcie sterownika FX3U z kilkoma modułami rozszerzenia, w tym z płytką modułu (tzw. adaptorem) 232-BD (złącze 9-stykowe; standard RS-232) i modułem 485-ADP ze złączem zaciskowym z sygnałami interfejsu RS-485.



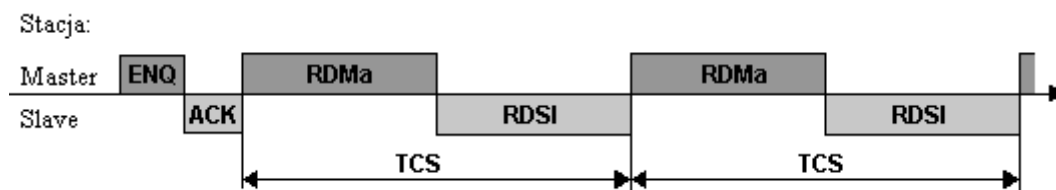
Rys. 1. Widok sterownika FX3U z komunikacyjnymi modułami rozszerzeń

Fig. 1. The view of the FX3U controller with communication extension modules

1.1. Sieci wbudowane

Sieci „1:1” i „N:N”, wbudowane w oprogramowanie systemowe sterownika FX zostały bardzo wyczerpująco przedstawione w pozycjach literaturowych [1, 2, 5]. Ich niezaprzeczalną zaletą jest bardzo proste uruchomienie, za pomocą aktywacji kilku markerów systemowych lub konfiguracji kilku rejestrów systemowych. Obie sieci działają według zasady Master-Slave. Na podstawie rozpoznania protokołu tych sieci [1] określono precyzyjne zasady

działania każdej z nich. Na rys. 2 przedstawiono wykres czasowy nadawania ramek protokołu sieci „1:1”, z oznaczeniem ich funkcji w okresie cyklu sieci TCS.

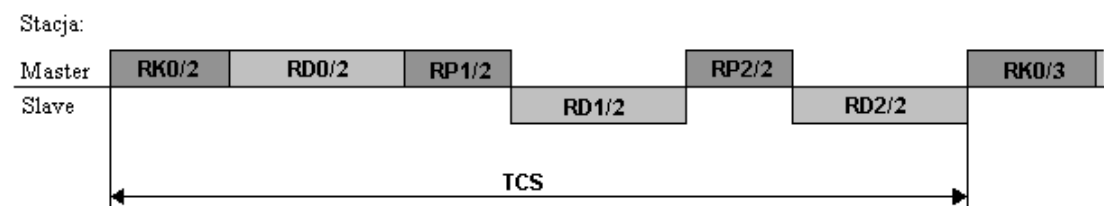


Rys. 2. Wykres czasowy nadawania ramek protokołu sieci „1:1”

Fig. 2. The time chart of frames transmission protocol for network “1:1”

Jak w każdej sieci Master-Slave inicjatywę podejmuje stacja Master wysyłając znak zapytania ENQ (ang. *enquire*, kod ze zbioru ASCII) do stacji Slave, czy ta jest gotowa do przekazu danych. Jeśli stacja Slave jest gotowa, to potwierdza ten stan nadaniem znaku potwierdzenia ACK (ang. *acknowledge*). Po tym wstępie stacja Master nadaje swoją ramkę danych RDMA, w odpowiedzi, na którą stacja Slave nadaje swoją ramkę danych RDSI. Następnie cykl się powtarza.

Cykl pracy sieci „N:N” jest bardziej złożony, ze względu na możliwie większą liczbę stacji sieciowych. Wykres czasowy jednego cyklu ramek sieci trzystacyjnej przedstawiono na rys. 3. Zastosowano tu – typowe dla struktury wielopunktu liniowego – wywołanie ramką polecenia RPi/k stacji Slave przez stację Master, w efekcie czego stacja Slave odpowiada swoją ramką danych RD_i/k. Liczba „i” oznacza adres stacji w ramce, liczba „k” przekazuje kolejny nr cyklu. Każdy cykl transferu sieciowego rozpoczyna się rozkazową ramką konfiguracyjną RKi/k. W ramce tej przekazywane są 4 parametry [1] dla stacji Slave w postaci: liczba stacji w sieci, wariant (zakres) transferu danych, liczba braku odpowiedzi danej stacji dla ustawienia markera awarii z daną stacją i maksymalny czas zwłoki nadania odpowiedzi każdej stacji sieciowej po otrzymaniu ramki rozkazowej.



Rys. 3. Wykres czasowy nadawania ramek protokołu sieci „N:N”

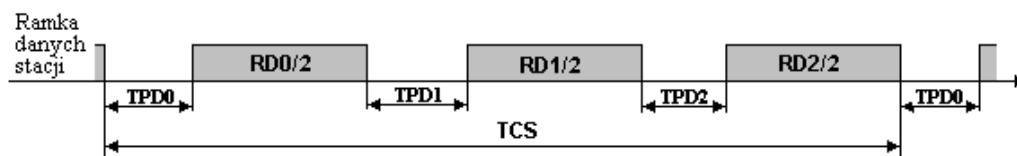
Fig. 3. The time chart of frames transmission protocol for network “N:N”

Sieć „N:N”, analogicznie do sieci „1:1”, pracuje według pełnego wariantu odwzorowania danych sieciowych (przekazywanych w ramach danych) w każdej stacji sieciowej. Odbywa się to na zasadzie przechwytywania danej ramki odpowiedzi przez każdą stację sieciową, co przy zastosowaniu łącza RS485 jest możliwe. Stąd też stacja Master nadaje również swoją

ramkę danych RD0/k, jako niby odpowiedź na ramkę konfiguracyjną RK0/k. Ramka RD0/k jest przeznaczona dla uzupełnienia odwzorowania w pozostałych stacjach sieciowych Slave.

1.2. Koncepcja sieci „K:K”

Idea i zasada działania sieci „K:K” zostały wyczerpująco przedstawione w [2]. Generalnie, na bazie dwóch wyżej wymienionych protokołów, zaproponowano tę nową sieć, tworzoną przez moduł programu aplikacyjnego. Protokół tej sieci został uproszczony do transferu wyłącznie ramek odpowiedzi, nazywanych tu ramkami danych RD_i. Został on przedstawiony na rys. 4.



Rys. 4. Wykres czasowy nadawania ramek protokołu sieci „K:K”

Fig. 4. The time chart of frames transmission protocol for network “K:K”

Idea protokołu wywodzi się z sieci „1:1”, gdzie ramka danych stacji Master inicjuje nadanie ramki stacji Slave. Z tej reguły wynika fakt, że nawet w systemie Master-Slave nie musi być wymagane nadanie specjalnej ramki rozkazowej, aby uzyskać odpowiedź z danej stacji. Rozkazem tym może być odbiór ramki danych innej stacji, w której adres nadawcy ramki wyznacza numer (adres) kolejnego nadawcy. Stąd też każda stacja przed nadaniem swojej RD_i/k potrzebuje określonego czasu przygotowania danych TPD_i, przeznaczonego na analizę poprzednio odebranej ramki i podjęcia decyzji o nadaniu własnej.

2. Podstawy programowe sieci „K:K”

Sprawdzenie koncepcji sieci „K:K” zrealizowano na tym samym sprzęcie, co omówione krótko sieci źródłowe tej idei, a co za tym idzie, w tym samym środowisku programistycznym. Założono, że programowy moduł realizujący zadania sieci dla danej stacji sieciowej ma być modułem uniwersalnym, konfigurowalnym automatycznie na podstawie podania wartości kilku parametrów konfiguracyjnych. Do tych zmiennych parametrów należą:

- liczba stacji sieciowych **K**, dopuszczająca wartość $K > N_{max}=8$,
- adres stacji sieciowej modułu **i**: $0 \leq i \leq K - 1$,
- liczba **L** słów 16-bitowych w ramce danych RD_i,
- szybkość transmisji sieciowej **V_{tr}** w jednostkach Bit/s,

- jednostkowa zwłoka czasowa **JZC**, konieczna do wyliczenia zwłoki czasowej **ZCi** nadania własnej ramki przez stację o adresie **i** (zależna od V_{tr}),
- czas ciszy w kanale transmisyjnym **Tci**, po upływie którego następuje zakończenie odbioru niepełnej ramki (ang. *time out*).

Do stałych parametrów sieci należy przyjęta struktura ramki danych RDi. Jest ona wynikiem wielu kompromisowych założeń, które rzutują na komplikację procesów jej tworzenia, a także jej dekodowania. Ponieważ oba procesy mają być realizowane drogą programu aplikacyjnego, stąd wymagane jest aby były one krótkie i przejrzyste, a także możliwie szybko realizowane. Struktura ta powinna przewidywać także możliwość diagnostyki sieci. Wariant protokołu spełniający powyższe założenia, dla $K_{max} = 16$ ($i_{max} = 15$) stacji sieciowych przedstawiono na rys. 5.

Nagłówek lub K D	Atest 0000 Adiag	Adr. „i” „i”	Słowo nr 0 16bitów	Słowo nr 1 16bitów	...	Słowo nr (L-1) 16bitów	CKR – ciąg kontr. ramki 16bitów
------------------------	------------------------	--------------------	-----------------------	-----------------------	-----	---------------------------	------------------------------------

Rys. 5. Struktura ramki protokołu sieci „K:K”

Fig. 5. The structure of the frame of the protocol of the network “K:K”

Założono, że ramka będzie się składała z **L+2** słów 16-bitowych, w której pierwsze słowo będzie słowem nagłówka, zawierającym bajt kodu nagłówka (znak K lub D), sieciowy adres stacji **Adr=i** oraz sieciowy adres stacji testowanej **Atest=Adiag** dla przeprowadzenia diagnostyki zdalnej ze stacji o adresie **Adr**. Znak nagłówka „K” (kod ASCII) jest przewidziany do normalnej pracy sieci, znak nagłówka „D” jest przewidziany do diagnostyki stacji o adresie **Adiag**. Ponieważ w rozpatrywanej sieci nie ma stacji Master, z której zazwyczaj taką diagnostykę się przeprowadza, stąd przewidziano tu możliwość przeprowadzenia diagnostyki danej stacji o adresie **Adiag**, z dowolnej stacji o adresie **Adr**. Diagnostyka jest dwuwariantowa i polega na wyłączeniu pracy w sieci stacji o adresie **Adiag** lub pozostawieniu pracy w sieci tylko tej stacji, a wyłączeniu wszystkich pozostałych.

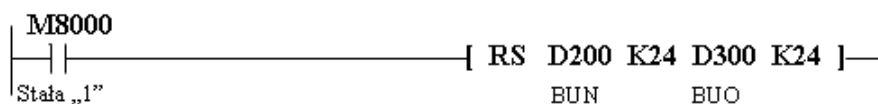
2.1. Programowa konfiguracja transmisji

Przygotowanie transmisji poprzez „lewostronny” port szeregowy wymaga inicjalizacyjnego zadeklarowania następujących parametrów:

- określenia w rejestrze systemowym **D8120** parametrów asynchronicznej ramki danych w postaci liczby bitów danych (7 lub 8), liczby bitów stopu, rodzaju bitu kontrolnego oraz szybkości transmisji,
- deklaracji w markerze systemowym **M8161**, czy dane do transmisji ramki sieciowej zajmują całe słowo 16-bitowe ($M8161 = ON$) czy tylko młodszą jego 8-bitową część ($M8161 = OFF$),

- deklaracji w rejestrze systemowym **D8129** wartości czasu ciszy w kanale transmisyjnym TCi.

Powyższe deklaracje dotyczą pierwszego kanału komunikacyjnego, zarządzanego instrukcją RS. Zarządzanie drugiego kanału może nastąpić instrukcją RS2 ze systemowymi rejestrami konfiguracyjnymi, zaczynającymi się od numeru 8400. Postać instrukcji RS w zapisie drabinkowym przedstawiono na rys. 6.



Rys. 6. Instrukcja aktywacji portu szeregowego

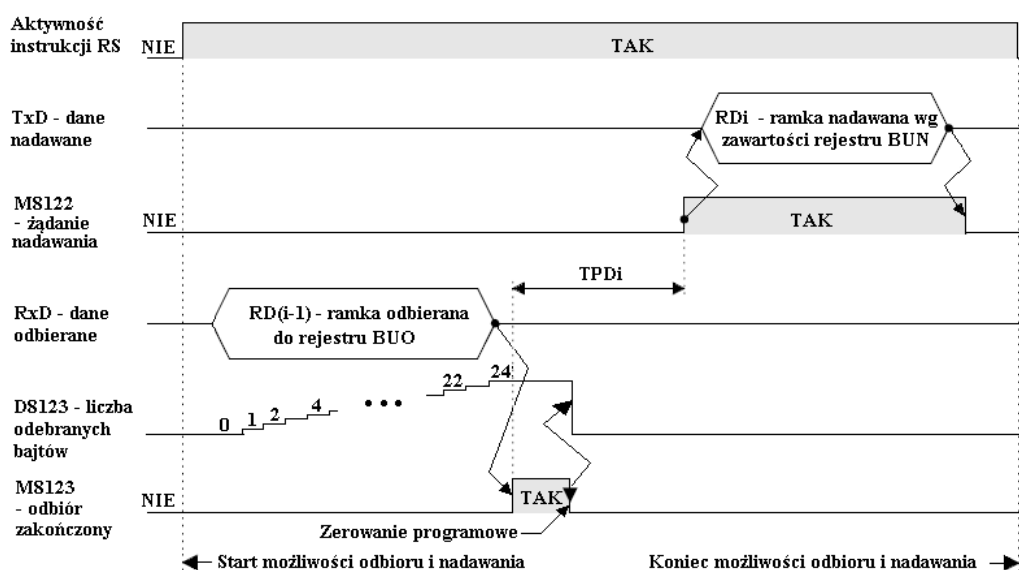
Fig. 6. The activation instruction of the serial port

Instrukcja zawiera cztery operandy, z których dwa pierwsze dotyczą bufora nadawczego **BUN**, gdzie D200 jest pierwszym rejestrem bufora, a K24 liczbą jego bajtów (K – liczba w systemie dziesiętnym). Analogicznie, bufor odbiorczy **BUO** deklarowany jest także pierwszym rejestrem bufora D300 i liczbą jego bajtów K24. Aby instrukcja działała bez przerwy, to musi być stale aktywowana, np. tak, jak na rys. 6 markerem M8000.

2.2. Programowa obsługa transmisji

Z instrukcją RS związanych jest wiele rejestrów i markerów systemowych z zakresu numeracyjnego 8120 – 8129. Na rys. 7 przedstawiono wykres czasowy działania wybranych urządzeń tego zakresu, podczas nadawania i odbioru ramki. Nadawanie ramki inicjowane jest markerem żądania nadawania M8122, zakończenie odbioru ramki sygnalizowane jest markerem M80123. Rejestr D8123 zawiera liczbę odebranych znaków przed zakończeniem odbioru, umożliwiającą stwierdzenie rozpoczęcia odbioru ramki (jej nadawania przez stację sieciową) przed faktycznym zakończeniem tego procesu.

Cechą szczególną instrukcji RS jest to, że została ona zaprojektowana do odbioru ramki o stałej długości, którą to długość deklaruje się w czwartym operandzie instrukcji RS. Dopiero po odebraniu liczby zadeklarowanych tu bajtów (lub znaków), marker M8123 przechodzi w stan TAK. W przypadku odebrania mniejszej liczby bajtów (np. odebrano tylko część ramki), zakończenie odbioru wyznaczane jest wartością czasu ciszy TC w kanale odbiorczym (ang. *time out*), którego wartość w jednostkach 10 ms ustawia się w rejestrze systemowym D8129. Po upływie tego czasu od odbioru ostatniego bajtu ustawiany jest marker M8129, który z kolei ustawia marker zakończenia odbioru M8123, analogicznie do rys. 7.



Rys. 7. Stan urządzeń i sygnałów sterownika podczas transmisji aktywowanej instrukcją RS
 Fig. 7. The state of the equipment and signals of the controller during the transmission initiated by the RS instruction

3. Moduł programowy sieci „K:K”

Dla weryfikacji założeń sieci „K:K” zaprojektowano moduł programowy na podstawie założeń z podrozdziału 2 dla następujących wartości parametrów: $K_{max} = 16$ stacji, $L = 10$ słów danych, $V_{tr} = 19200$ bit/s, $JZC = 20$ ms (zakładana max. suma czasów realizacji pętli programowej i transmisji ramki) i $TC = 10$ ms. Dalej zostaną omówione wybrane sekcje funkcjonalne modułu sieciowego.

3.1. Sekcja dekodowania ramki

Zanim zostanie podjęta decyzja o zarejestrowaniu danych odebranej ramki danych RDj stacji o adresie „j” i ewentualnym przejściu do nadania ramki własnej, to ramka ta musi być sprawdzona pod kątem jej poprawności. Na kontrolę tę (inicjowaną stanem TAK markera M8123) składają się (w kolejności) następujące czynności weryfikacyjne:

- czy liczba odebranych danych w D8123 wynosi $n = L + 2$,
- czy ramkę odebrano bez błędów, na podstawie ciągu kontrolnego ramki CKR,
- czy odebrana ramka posiada nagłówek sieciowy „K” czy diagnostyczny „D”,
- czy odebrany w ramce RDj adres stacji „j” jest w zakresie: $0 \leq j \leq K - 1$.

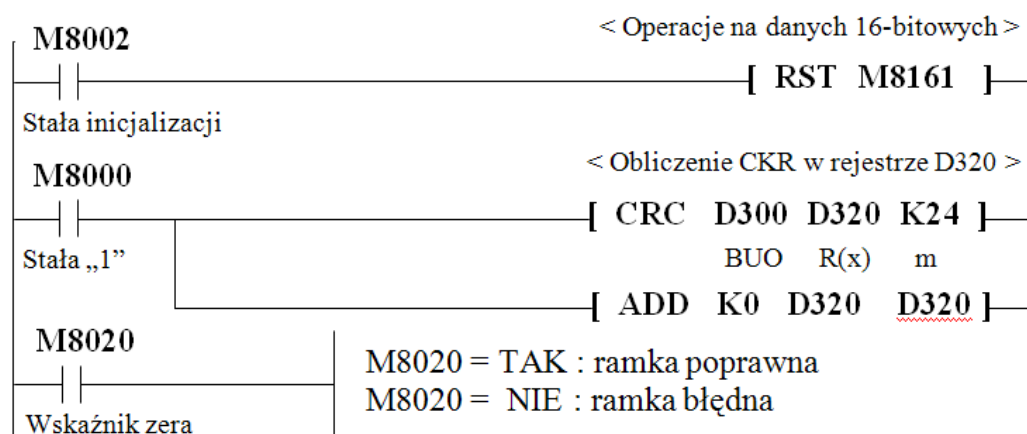
Efektom pozytywnego wyniku powyższych testów jest wpisanie stanu „1” na pozycji „j” w 16-bitowym rejestrze statusu odbioru RSO. Każda pozytywna weryfikacja odebranej ramki danych kolejnej stacji zapisuje odpowiadającą jej pozycję w tym słowie. Nadanie własnej

ramki RDi powoduje zapisanie pozycji nr „i” w RSO, przepisanie RSO do 16-bitowego rejestru stanu cyklu RSC i wyzerowaniu RSO. W ten sposób w RSC jest informacja o tym, które obrazy danych stacji sieciowych są aktualne, a które nie. RSC jest głównym rejestrem statusem sieci w każdej stacji.

3.2. Weryfikacja nadmiaru kodowego

Ostatnim słowem ramki protokołu sieci „K:K” (rys. 5) jest 16-bitowe słowo ciągu kontrolnego ramki, oznaczane skrótem CKR. Słowo to jest kodowym nadmiarem informacyjnym, wprowadzonym celowo w stacji nadawczej, w celu uzyskania możliwości weryfikacji poprawności ramki (braku przekłamań) w stacji odbiorczej. Z tradycyjnie stosowanych dwóch metod kodowania (kod iterowany dla struktury tablicowej i kod wielomianowy dla struktury ciągu bitów) zastosowano metodę kodowania wielomianowego.

Sposób ten polega na potraktowaniu tablicowej struktury danych jako ciągu szeregowego kolejnych słów tablicy. Ciąg ten dalej traktowany jest jako ciąg współczynników binarnych wielomianu stopnia $m=L \times 16 - 1$. Na wielomianie tym dokonuje się operacji dzielenia przez tzw. wielomian generujący $G(x)$, a reszta z tego dzielenia tworzy CKR. W sterownikach FX3U wprowadzono instrukcję dzielenia wielomianowego o symbolu CRC (cykliczna kontrola nadmiarowa; ang. *cyclic redundancy check*) przez wielomian generujący CRC_{16} stopnia $n=16$, w postaci $G_{CRC} = x^{16} + x^{15} + x^2 + 1$, gdzie symbol „+” w tym zapisie oznacza operator sumy modulo 2. Na rys. 8 przedstawiono fragment programu dekodowania ramki z instrukcją CRC z operandami, wykonywaną na słowach 16-bitowych.



Rys. 8. Sprawdzenie poprawności ramki transmisyjnej odebranej w buforze odbiorczym BUO
Fig. 8. The verification check of the transmission frame received in the receive buffer BUO

Instrukcja CRC oblicza w rejestrze D320 resztę z dzielenia ciągów binarnych zawartych w rejestrach bufora odbiorczego BUO przez wielomian $G(x)$. Stała K24 oznacza liczbę 24 bajtów bufora. Jeśli odebrana ramka w BUO jest poprawna, to jest ona podzielna bez reszty

przez wielomian $G(x)$, czyli w rejestrze reszty D320 wartość reszty $R(x) = 0$. Wartość tę sprawdza instrukcja dodawania ADD, której zadaniem jest ustawienie wskaźnika zera wyniku operacji w postaci markera M8020. Stan tego markera decyduje o dalszym przebiegu programu.

Proces kodowania ramki nadawanej wykorzystuje także instrukcję CRC. W tym przypadku jej zadaniem jest obliczenie reszty $R(x)$ z ciągu 11 słów początkowych ramki i wpisanie tej reszty do rejestru przenoszącego CKR.

Zabezpieczenie wielomianowe ramki za pomocą standardowego wielomianu CRC_{16} daje taką samą odległość Hamminga $d_H = 4$, jak kod iterowany. Odporność na błędy seryjne jest większa i wynosi $\gamma_S = n$. Oznacza to, że kod wykrywa wszystkie paczki błędów do długości 16 włącznie.

3.3. Token adresowo-czasowy

Ponieważ działanie sieci oparto na przechwytywaniu danych, czyli odbiorze ramki danych RDi i-tej stacji przez wszystkie pozostałe stacje, stąd adres „i” w odebranej ramce jest traktowany w każdej stacji jako **token (znacznik) adresowo-czasowy**. Wyznacza on stację o adresie $Adr = i+1$ (modulo K) do bezzwłocznego nadania swojej ramki danych. Natomiast w każdej z pozostałych stacji token wyznacza wartość zwłoki czasowej ZC_j , po upływie której stacja nr „j” ma nadać swoją ramkę, jeśli stacja (stacje) o młodszym adresie z jakiejś przyczyny nie nadałyby swojej ramki. Wartość $ZC_j = k \times JZC$, gdzie „k” jest odległością adresową pomiędzy stacją nr „j” a „i”, przyjmującą wartości: $k = j - i$ dla $j > i$, oraz $k = K - 1 + j - i$ dla $j < i$ [2]. Ten token czasowy, w postaci każdorazowo obliczonej zwłoki czasowej ZC_j , wznawia transmisję w sieci po przerwie z jakąś stacją i zapobiega konfliktowi dostępu do kanału w sieci. Dla nowo włączonej stacji do sieci wartość $k = K - 1$. Dla przetestowania poprawności działania tokena czasowego przewidziano dwie opcje diagnostyki sieci, sygnalizowane w podrozdziale 2.

3.4. Czas transmisji stacji sieciowej

Dla parametrów sieci przyjętych w podrozdziale 3 napisano program aplikacyjny tej sieci dla $K = 8$ stacji sieciowych. Sprawdzono z wynikiem pozytywnym działanie tokena adresowo-czasowego, wielomianowego kodowania nadmiarowego oraz aktualizację słowa statusowego cyklu. Na rys. 9 przedstawiono czasową rejestrację ramek sieciowych dla przypadku nadających wszystkich 8 stacji ze zwłoką własną 10 ms (12,5 ms trwa transmisja ramki).

time:	hex:
18:46:30.177	07 4b 00 00 00 00 00 00 00 00 00 00 00 00 00 07 ff 00 00 00 00 06 e4
18:46:30.200	00 4b 00 ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 33 17
18:46:30.223	01 4b 00 00 01 ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 44 70
18:46:30.247	02 4b 00 00 00 00 02 ff 00 00 00 00 00 00 00 00 00 00 00 00 00 31 2b
18:46:30.271	03 4b 00 00 00 00 00 00 03 ff 00 00 00 00 00 00 00 00 00 00 00 72 f0
18:46:30.294	04 4b 00 00 00 00 00 00 04 ff 00 00 00 00 00 00 00 00 00 00 00 06 d5
18:46:30.317	05 4b 00 00 00 00 00 00 05 ff 00 00 00 00 00 00 00 00 00 00 00 c6 e7
18:46:30.341	06 4b 00 00 00 00 00 00 06 ff 00 00 00 00 00 00 00 00 00 00 00 06 e4
18:46:30.364	07 4b 00 00 00 00 00 00 07 ff 00 00 00 00 00 00 00 00 00 00 00 06 e4
18:46:30.388	00 4b 00 ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 33 17

Rys. 9. Rejestracja czasowa ramek danych RD w sieci „K:K” dla K=8

Fig. 9. The RD data frames time registration in the network “K: K” for K = 8

4. Podsumowanie

Uruchomienie modułu sieciowego „K:K”, jako stałej części programu aplikacyjnego dla danej stacji potwierdziło jej podstawowe założenie, że do inicjowania transmisji danych sieciowych nie jest konieczna stacja Master, jako stacja wyznaczająca reguły transmisji w sieci. Poprzez nieobecność ramek rozkazowych jest możliwe osiągnięcie większej efektywnej szybkości wymiany danych stacyjnych, co było jednym z głównych założeń sieci „K:K”.

W artykule, ze względu na narzucone ograniczenie jego rozmiaru, poruszone zagadnienia omówiono bardzo krótko, a wiele z nich pominięto zupełnie. Do tych ostatnich należą:

- wpływ czasu cyklu sterownika na czas ciszy TC i jednostkową zwłokę czasową JZC,
- możliwości wprowadzenia zmiennej liczby danych w ramce danych RD_i,
- głębsza analiza konfliktów w sieci z wnioskami ich zażegnania,
- uwarunkowania i wyniki zastosowania wersji diagnostycznej modułu sieciowego.

Z istnienia powyższych zagadnień wynika, że tematyka modułu programowego sieci „K:K” wymaga jeszcze dalszych badań.

BIBLIOGRAFIA

1. Mielcarek R.: Protokoły sieci wbudowanych sterowników kompaktowych serii FX firmy Mitsubishi Electric. Techniczne i teoretyczne aspekty współczesnych sieci komputerowych, WKŁ, Warszawa 2009, s. 185÷196.
2. Mielcarek R.: Dydaktyczna sieć programowa „K:K”. Współczesna Problematyka Sieci Komputerowych, WKŁ, Warszawa 2010, s. 173÷184.
3. Mitsubishi Electric: FX3U/FX3UC Programming Manual. M. No. JY997D16601, 7/2005.
4. Mitsubishi Electric: FX3U Base Units and Extension Bloks. M. No. JY997D16501, 6/2007.

5. Mitsubishi Electric: FX1S/FX1N/FX2N/FX3U Interface Modules. M. No. JY997D16901, 11/2007.
6. Mitsubishi Electric: FX3G User`s Manual. M. No. JY997D31301, 11/2008.
7. Mitsubishi Electric: Księga automatyzacji – Świat rozwiązań. 2010/2011.
8. Strona internetowa: www.mitsubishi-automation.pl.

Recenzenci: Prof. dr hab. inż. Stanisław Kozielski
Dr hab. inż. Andrzej Kwiecień, prof. Pol. Śląskiej

Wpłynęło do Redakcji 13 marca 2011 r.

Abstract

This paper presents the idea of a software module of the industrial network named “K:K”. The module was written for the FX series PLCs from Mitsubishi Electric. The fastest member of this series, FX3U, was shown in Figure 1. The presence of additional serial port on the controller allows to launch one of the two industrial networks called “1:1” and “N:N”. The network software is built-in the system driver. These networks are discussed briefly in subsection 1.1. The structure of the network protocols is shown in Fig. 2 and Fig. 3. The principle of operation of those networks defined the concept of network programming “K:K”. Its protocol is shown in Figure 4. Next, the core curriculum of the “K:K” was discussed. At Fig. 5 the protocol was shown, and further, on Fig. 6, the software instruction RS with the frame transmission protocol parameters. The serial port support is done according to the guidelines from Fig. 7. The next section, No. 3, discusses the assumptions of the network software module. It discusses four, the following issues: stages of frame decoding, verification of redundant coding using the polynomial CRC16 (Fig. 8), address-time network token, obtained network time performance.

In Fig. 9 shows the registration time of the protocol frames, generated by the designed network module. The chapter ends with a brief summary.

Adres

Roman MIELCAREK: Politechnika Poznańska, Instytut Informatyki, ul. Piotrowo 2,
60-965 Poznań, Polska, roman.mielcarek@cs.put.poznan.pl