

Analiza istniejącego protokołu transmisji PPM2 danych i informacji w sieci CAN-BUS

Marek KANIEWSKI¹, Jan MALICKI²

Streszczenie

W artykule przedstawiono ocenę stosowanego protokołu PPM2 służącego na PKP do nadzorowania pracy obiektów energetycznych w szczególności wyłączników szybkich. Poza koleją ten protokół jest stosowany w tramwajach i trolejbusach, nie tylko do nadzoru pracy wyłączników szybkich, ale również przy nadzorze pracy oświetlenia lub ogrzewania rozjazdów w zimie. Przedstawiono wnioski ankiety przesłanej do dotychczasowych użytkowników protokołu PPM2 i na tej podstawie zaproponowano wprowadzenie niezbędnych, nowych funkcji w tym protokole.

Słowa kluczowe: podstacja trakcyjna, wyłącznik szybki, zdalne sterowania wyłącznikami szybkimi

1. Wstęp

Celem artykułu jest ocena przydatności protokołu PPM2, po kilkunastu latach jego stosowania, do obecnych potrzeb automatyki i zabezpieczeń obiektów energetyki trakcyjnej. Protokół PPM2 powstał w Instytucie Kolejnictwa (Centrum Naukowo-Techniczne Kolejnictwa) przy współpracy z przedsiębiorstwem Elester-PKP Sp. z o.o. i jest przeznaczony dla urządzeń korzystających z magistrali CAN-Bus w obiektach trakcji kolejowej. Potrzeba stworzenia takiego protokołu z wykorzystaniem do tego celu magistrali CAN-Bus wynikała z konieczności zapewnienia bardzo „ostrzych” wymagań czasowych w zakresie przesyłania wybranych informacji, niezbędnych do prawidłowej pracy zabezpieczeń. Protokół ten jest stosowany od blisko 20 lat i z powodu swoich zalet jest powszechnie stosowany również w innych dziedzinach trakcji elektrycznej, takich jak trakcja tramwajowa lub trolejbusowa. Wykorzystywany jest również w systemach sterowania oświetleniem i elektrycznego ogrzewania rozjazdów. Dzięki temu protokołowi jest możliwe tworzenie spójnych systemów urządzeń produkowanych przez różnych producentów.

2. Opis i ocena dotychczasowych rozwiązań

Obecnie powszechnie stosowana w obiektach zasilania trakcji elektrycznej magistrala CAN-Bus z protokołem PPM2 spełnia swoje zadania w podstawowym zakresie funkcji.

Rozwiązanie to zapewnia doskonały „arbitraż”³ w zakresie dostępu do magistrali użytkowników mających najważniejsze informacje do wysłania w danej chwili. Po odpowiednim wykonaniu i uruchomieniu, magistrala CAN-Bus nie naraża również problemów eksploatacyjnych. Protokół PPM2 wykorzystuje starszą wersję arbitrażu – 11-to bitową. Ponieważ 3 bity z 11 są wykorzystane do celów specjalnych, to maksymalna liczba urządzeń mogących mieć dostęp w magistrali nie może przekraczać 2^8-1 , tj. 255 sztuk. Z powodów ograniczonej obciążalności fizycznej sygnałów magistrali, współpraca takiej liczby sterowników nie jest możliwa bez stosowania regeneratorów sygnałów.

W ostatnich latach nowe konstrukcje zabezpieczeń, poza typowymi funkcjami związanymi z realizacją zadań zabezpieczeniowych i telemechaniki, są wyposażane w tzw. kanał inżynierski. Powoduje to konieczność wyposażania tych urządzeń w dostęp do standardowych sieci transmisji danych. Rozwiązania bazujące na sieci CAN-Bus są predestynowane do realizacji zadań czysto technicznych i mają istotne ograniczenia w zakresie szybkości jak i przesyłaniu dużych bloków informacji.

Z tych powodów nowe zabezpieczenia z reguły są wyposażane w porty umożliwiające dostęp do sieci Ethernet, niezależnie od podwójnej magistrali CAN-Bus. Powstaje sytuacja, że w obiekcie są realizowane trzy niezależne sieci. Kanał inżynierski stanie się prawdopodobnie trwałym i rozwojowym elementem, który już dzisiaj umożliwia między innymi:

¹ Mgr inż.; Instytut Kolejnictwa, Zakład Elektroenergetyki; e-mail: Kaniewski@ikolej.pl.

² Mgr inż.; e-mail: jan.malicki@wp.pl.

³ Priorytet w przesyłaniu danych wynikająca z ich ważności.

- dostęp do pełnej rejestracji zdarzeń występujących na obiektach, włącznie z możliwością śledzenia sekwencji zdarzeń, profili obciążeń oraz wykresów zwarciovych,
- podgląd wszelkich nastaw zabezpieczeń, możliwość ich korekty oraz podgląd historii zmian z możliwością ustalenia ich autorów,
- planowanie przeglądów, konserwacji i remontów z uwzględnieniem historii eksploatacji,
- przeprowadzenie prób zwarciovych z pełną rejestracją.

Celowym wydaje się rozważenie, czy w związku z tym, w przyszłości nie należałoby zastąpić wymienionych sieci jedną siecią typu Ethernet wykorzystującą światłowody na terenie obiektów i realizującą komplet funkcji. Stosowanie sieci światłowodowej pozwoli na pełną separację elektryczną poszczególnych zabezpieczeń. W takiej sieci można wydzielić sieci VPN dla telemechaniki i kanału inżynierskiego. Powinna ona pracować na obiekcie w ringu, co pozwoliłoby uniknąć tworzenia dwóch sieci jak w przypadku sieci CAN-Bus. Przy wykorzystaniu odpowiednio skonfigurowanych urządzeń, sieć pracująca z prędkością 1 Gbps jest w stanie zapewnić właściwą współpracę zabezpieczeń.

3. Ankietowanie istniejącego protokołu PPM2

Instytut Kolejnictwa przygotował listę firm, które w ostatnich latach produkowały i dostarczały na potrzeby PKP Energetyka Sp. z o.o. urządzenia współpracujące z magistralą CAN-Bus z wykorzystaniem protokołu PPM2 oraz wysłał do nich prośbę o zgłoszenie uwag w zakresie potrzeb związanych z uzupełnieniem lub modyfikacją wymienionego protokołu. W szczególności dotyczyło to uzupełnienia opisu protokołu PPM2 w zakresie informacji potrzebnych do pracy uzależnień wyłączników szybkich. Do współpracy zaproszono dziesięć firm, na prośbę Instytutu odpowiedziały cztery firmy, a ich uwagi w miarę możliwości uwzględniono w opracowaniu protokołu PPM2. Dotyczyły one warstwy fizycznej i warstwy logicznej protokołu. Z nadesłanych informacji wynika, że protokół PPM2 wraz z wykorzystywaną magistralą CAN-Bus, jest w dalszym ciągu rozwiązaniem spełniającym swoje podstawowe funkcje w obiektach zasilania

W trakcie prac nad uzupełnieniem i aktualizacją protokołu PPM2, producenci sprzętu sygnalizowali dodatkowe problemy wymagające pilnego rozwiązania w zakresie konwersji telegramów przesyłanych po magistrali CAN-Bus, związanych z pracą uzależnień wyłączników szybkich za pomocą telegramów przesyłanych do współpracujących obiektów po sieciach Ethernet i VPN. Protokół nazwany UPK2, został opracowany [1, 2] i powinien on zastąpić obecnie stosowany protokół UPK.

4. Protokół przekazywania informacji na magistrali CAN-BUS/RS485 – PPM2 V3.0

Producenci sprzętu nie zgłosili znaczących potrzeb w zakresie modyfikacji funkcjonalnej lub dodatkowych telegramów w protokole PPM2. Zgłoszone wnioski umieszczono w zaktualizowanym opisie protokołu oznaczonym jako wersja 3.0.

Pole arbitrażu⁴ ID wykorzystywane przez układy nadawczo-odbiorcze sieci zawiera 11 bitów. Informacje umieszczane w tym polu są niezbędne do właściwego działania sieci, a zwłaszcza ustalania priorytetów przesyłanych informacji. Pole arbitrażu podzielono na dwa bloki: priorytetu i klasy obiektu. Blok priorytetu stanowi trybitową starszą część pola arbitrażu (bity ID10..ID8), pozwalającą na ustalenie 8 priorytetów (od 0 do 7). Priorytety przydzielone telegramom zamieszczono w tablicy 1.

Tablica 1

Blok priorytetu pola arbitrażu

Kategoria telegramu	Priorytet (Pola ID10..ID8)	Uwagi
Rezerwa	000	
Synchronizacja czasu	001	Wysyłane co 1–2 s
Meldunki bardzo szybkie	010	Wysyłane przez 2,5 s; trzy pierwsze co 10 ms, następne co 100 ms
Polecenia i potwierdzenia	011	Wysyłane jednorazowo
Meldunki szybkie	100	Wysyłane jednorazowo
Meldunki cykliczne	101	Wysyłane co 0,5–1 s
Telegramy użytkownika	110	Wysyłane nie częściej niż co 200 ms
Dane, programy, nastawy Zapis / odczyt rejestrów	111	Całość wysyłana w sposób ciągły

Osiem młodszych bitów pola arbitrażu (pola ID7..ID0) określa kategorię (rodzaj) urządzenia nadającego telegram. Istnieje możliwość przydzielenia 240 numerów. Numery poszczególnych urządzeń dla sieci w podstacji trakcyjnej zamieszczono w tablicy 2. W przypadku innych zastosowań należy przydzielić urządzeniom numery w zależności od wymaganego priorytetu ich dostępu do sieci (urządzenia o niższym numerze mają wyższy priorytet).

⁴ Ciąg 11 bitów decydujący o priorytetach w przesyłaniu danych i informacji

Tablica 2

Blok kategorii urządzenia pola arbitrażu

Kategoria urządzeń	Numer binarnie (Pola ID7. ID0)	Numery urządzeń	
		szesnastkowo	dziesiętnie
Ochrona ziemnozwarciowa	od 0010.0000 do 0010.0001	20h-21h	032-033
Ochrona podnapięciowa	od 0010.0100 do 0010.0111	24h-27h	036-039
Transmisja/urządzenia uzależnień	od 0010.1000 do 0010.1111	28h-2fh	040-047
Urządzenia zabezpieczeń	od 0011.0000 do 0101.1111	30h-5fh	048-095
Zasilacze	od 0110.0000 do 0111.1100	60h-7bh	096-123
Zasilacze zapasowe	od 0111.1101 do 0111.1111	7ch-7fh	124-127
Zespoły prostownikowe	od 1010.0000 do 1010.0111	a0h-a7h	160-167
Urządzenia różne	od 1010.1000 do 1010.1111	a8h-afh	168-175
Odłączniki	od 1011.0000 do 1011.1111	b0h-bfh	176-191
Linie potrzeb nietrakcyjnych	od 1100.1000 do 1100.1111	c8h-cfh	200-207
Linie zasilające	od 1101.0000 do 1101.0011	d0h-d3h	208-211
Zdalne sterowanie	od 1101.0100 do 1101.0101	d4h-d5h	212-213
Urządzenia pomocnicze	od 1101.0110 do 1101.1111	d6h-dfh	214-223
Terminal	od 1110.0000 do 1110.0001	e0h-e1h	224-225
Testery diagnostyczno-serwisowe	od 1110.0010 do 1110.0100	e2h-e4h	226-228
Urządzenia pomocnicze	od 1110.0101 do 1110.1111	e5h-efh	229-239
Numery zabronione	od 1111.0000 do 1111.1111	f0h-ffh	240-255

Tablica 3
Struktury pola danych

Typ telegramu	Pole typu	Liczba bajtów
Meldunki		
Meldunek zwykły	1	1+4
Meldunek ze znacznikami czasu	4	1+7
Meldunek rozszerzony	18	1+6
Polecenia		
Polecenie	5	1+4
Potwierdzenia		
Potwierdzenie	6	1+4
Przesyłanie programów, nastaw, danych		
Programowanie kanału transmisji danych	13	1+3,5,6
Status urządzenia transmisji danych	14	1+7
Transmisja danych – zapis	15	1+4,5,7
Transmisja danych – odczyt	16	1+3,4,6
Synchronizacja czasu		
Synchronizacja czasu	17	1+7
Zapis / odczyt rejestrów		
Zapis rejestru CHAR	19	1+4
Polecenie odczytu rejestru CHAR	20	1+3
Potwierdzenie zapisu lub odczyt rejestru CHAR	21	1+4
Zapis rejestru INT	22	1+5
Polecenie odczytu rejestru INT	23	1+3
Potwierdzenie zapisu lub odczyt rejestru INT	24	1+5
Zapis rejestru LONG	25	1+7
Polecenie odczytu rejestru LONG	26	1+3
Potwierdzenie zapisu lub odczyt rejestru LONG	27	1+7
Telegramy użytkownika		
Stany wewnętrzne	32	1+3,4,5,6,7

Opis struktury pola danych

Pole danych może zawierać od 0 do 8 bajtów danych. Struktury pola danych są częściowo narzucone, a częściowo mogą być zmieniane przez twórców poszczególnych urządzeń występujących na podstacjach trakcyjnych. W tablicy 3 zamieszczono przydzielone typy telegramów i odpowiadające im wartości pola typu telegramu.

Meldunki

Meldunki służą do przesyłania informacji z urządzeń. Każde urządzenie może wysłać maksymalnie do 256 serii meldunkowych, z których każda zawiera 16 bitów danych (tablica 4 i 5).

Tablica 4

Meldunki – struktura telegramów typ 1-4

Bajt danych	Opis	Wartość	Uwagi
1	Typ telegramu	1	Meldunek zwykły
		4	Meldunek ze znacznikami czasu
2	Nadawca logiczny	1..239	Według tablicy 2
3	Numer serii	0..255	
4	Młodszy bajt danych	0..255	
5	Starszy bajt danych	0..255	
Dodatkowe pola dla typu telegramu =4 (Meldunki ze znacznikami czasu)			
6	Minuta	0..59	
7	Sekunda	0..59	
8	Dziesiątki milisekund	0..99	

Tablica 5

Meldunki – struktura telegramów typ 18

Bajt danych	Opis	Wartość	Uwagi
1	Typ telegramu	18	Meldunek rozszerzony
2	Nadawca logiczny	1..239	Według tablicy 2
3	Numer serii	0..255	
4	Młodszy bajt danych 1 serii pomiarowej	0..255	
5	Starszy bajt danych 1 serii pomiarowej	0..255	
6	Młodszy bajt danych 2 serii pomiarowej	0..255	
7	Starszy bajt danych 2 serii pomiarowej	0..255	

Polecenia tradycyjne i potwierdzenia

Wykonanie polecenia przebiega w dwóch etapach. W pierwszym etapie urządzenie zapamiętuje polecenie normalne (pole typu=5 z rodzajem polecenia=4) bez wykonywania go i odsyła potwierdzenie odbioru (pole typu=6 z rodzajem potwierdzenia=4) z zawartością właśnie odebranego kodu polecenia. Następnie stanowisko oczekuje na polecenie wykonawcze (pole typu=5 z rodzajem polecenia=3) zawierające kod polecenia wykonawczego, po odebraniu którego wykonuje zadaną operację i po jej ukończeniu znów potwierdza wykonanie (pole typu 6). Jeśli wykonanie przebiegło poprawnie, jest wysyłane potwierdzenie prawidłowego wykonania polecenia (rodzaj potwierdzenia=3) z kodem polecenia wykonawczego. W przeciwnym razie jest wysyłane potwierdzenie błędnego wykonania polecenia (rodzaj potwierdzenia=5) z kodem błędu. W przypadku, gdy od odebrania polecenia normalnego przez 20 sekund nie odebrano polecenia wykonawczego, rozpoczęte wykonywanie polecenia jest anulowane. Opis struktury bajtów poleceń podano w tablicy 6, a strukturę bajtów w tablicy 7.

Tablica 6

Polecenia – struktura telegramów

Bajt danych	Opis	Wartość	Uwagi
1	Typ telegramu	5	Polecenie
2	Odbiorca logiczny	0..239	0 – wszyscy 1..239 – według tablicy 2
3	Rodzaj polecenia	0..4	0 – reset urządzenia 1 – kontrolne wspólne 2 – kontrolne indywidualne 3 – wykonawcze 4 – normalne (sterujące)
4	Młodszy bajt kodu polecenia	0..255	Kodowane według tablicy poleceń
5	Starszy bajt kodu polecenia	0..255	

⁵ Zgodne wartości świadczą o tym, że wszystkie dane dotarły do adresatów.

Tablica 7

Potwierdzenia – struktura telegramów

Bajt danych	Opis	Wartość	Uwagi
1	Typ telegramu	6	Potwierdzenie
2	Nadawca logiczny	1..239	Według tablicy 2
3	Rodzaj potwierdzenia	0..5	0 – potwierdzenie resetu urządzenia 1 – potwierdzenie prawidłowego wykonania polecenia kontrolnego wspólnego 2 – potwierdzenie prawidłowego wykonania polecenia kontrolnego indywidualnego 3 – potwierdzenie prawidłowego wykonania polecenia wykonawczego 4 – potwierdzenie prawidłowego wykonania polecenia normalnego (sterującego) 5 – potwierdzenie błędnego wykonania polecenia
4	Młodszy bajt kodu potwierdzenia	0..255	Kodowane według tablicy poleceń lub kod błędu
5	Starszy bajt kodu potwierdzenia	0..255	

Programy, nastawy, dane

Przesyłanie programów, nastaw, danych odbywa się z wykorzystaniem tzw. kanałów transmisji danych. Możliwa jest jednoczesna transmisja danych do / z wielu urządzeń. Przesyłanie danych do grupy urządzeń (lub pojedynczego urządzenia) przebiega następująco:

- urządzenie inicjujące transmisję (nadawca) otwiera kanał do zapisu w urządzeniach, do których musi przesłać dane (odbiorcy),
- odbiorcy odpowiadają, podając swój status,
- nadawca ustawia adres bazowy,
- odbiorcy odpowiadają, podając swój status,
- nadawca przesyła dane do adresatów, którzy nie potwierdzają kolejnych telegramów zawierających dane,
- nadawca żąda podania statusu od urządzeń, do których transmitował dane,
- odbiorcy odpowiadają, podaniem swoich statusów,
- nadawca porównuje liczbę wysłanych bajtów danych z liczbą bajtów, które odebrali adresaci (informacja zawarta w polu statusu⁵),
- nadawca może podać nowy adres bazowy i zapisywać dane do innego obszaru,
- nadawca zamyka kanał transmisji,
- odbiorcy potwierdzają zamknięcie kanału podając swój status.

Operacja odczytywania danych z grupy urządzeń lub pojedynczego urządzenia przebiega następująco:

- urządzenie inicjujące transmisję (odbiorca) otwiera kanał do odczytu w urządzeniach, z których chce odczytać dane (nadawcy),
- nadawcy odpowiadają podając swój status,
- odbiorca przesyła do nadawców adres bazowy, od którego transmitowane będą dane,
- nadawcy potwierdzają odsyłając swój status,
- odbiorca przesyła polecenie odczytu danych, podając przy tym długość bloku danych do transmisji,
- nadawcy przesyłają swój status, następnie wysyłają żadaną ilość danych,
- odbiorca porównuje liczbę odebranych bajtów danych od poszczególnych urządzeń z liczbą bajtów jaką zażądał; zgodne wartości świadczą o tym, że wszystkie dane zostały odebrane,
- odbiorca może podać nowy adres bazowy i odczytać dane z innego obszaru,
- po zakończeniu transmisji odbiorca zamyka kanał,
- nadawcy potwierdzają zamknięcie kanału podając swój status.

Struktury telegramów rozkazów sterujących, statusu urządzeń transmisji danych, zapisu danych i odczytu danych podano w tablicach 8–11.

Tablica 8

Programy, nastawy, dane: rozkazy sterujące – struktura telegramów

Bajt danych	Opis	Wartość	Uwagi
1	Typ telegramu	13	Programowanie kanału
2	Adresat od	0..239	Dane dla urządzeń o adresach z zakresu: Adresat od – Adresat do
3	Adresat do	0..239	
4	Kod operacji	0	Żądanie podania statusu
		1	Otwarcie kanału do zapisu
		2	Otwarcie kanału do odczytu
		3	Nowy adres bazowy
		4	Polecenie odczytu danych
		5	Zamknięcie kanału transmisji
		6	Zeruj licznik przesłanych bajtów
Dodatkowe pola dla kodu operacji =3 (Nowy adres bazowy)			
5	Adres bazowy (trzy najstarsze bajty)	0..255	Młodszy bajt
6		0..255	
7		0.255	Starszy bajt
Dodatkowe pola dla kodu operacji=4 (Polecenie odczytu danych)			
5	Adres początkowy względny	0..255	Przesunięcie od adresu bazowego
6	Adres końcowy względny	0..255	Przesunięcie od adresu bazowego

Tablica 9

Programy, nastawy, dane: status urządzenia transmisji danych – struktura telegramów

Bajt danych	Opis	Wartość	Uwagi
1	Typ telegramu	14	Status transmisji danych
2	Numer urządzenia żądającego dane	0..239	
3	Potwierdzenie	0	Urządzenie otwarte do zapisu
		1	Urządzenie otwarte do odczytu
		2	Urządzenie zamknięte do odczytu i zapisu
		3	Błąd: adres nieprawidłowy
		4	Błąd: urządzenie chwilowo niedostępne
		5..255	Inne błędy
4	Adres bazowy (trzy najstarsze bajty)	0..255	Młodszy bajt
5		0..255	
6		0..255	Starszy bajt
7	Liczba przesłanych bajtów	0..255	Licznik przesłanych bajtów od 1 do 256
8		0..255	

Uwaga: status urządzenia transmisji danych wysyłany jest po każdym odebraniu polecenia „programowanie kanału” oraz polecenia „żądanie podania statusu”.

Tablica 10

Programy, nastawy, dane: zapis – struktura telegramów

Bajt danych	Opis	Wartość	Uwagi
1	Typ telegramu	15	Zapis danych
2	Adresat od	0..239	Dane dla urządzeń o adresach z zakresu: Adresat od – Adresat do
3	Adresat do	0..239	
4	Adres względny	0..255	Przesunięcie od adresu bazowego
Przesyłanie jednego bajta danych			
5	Bajt danych nr 1	0..255	
Przesyłanie dwóch bajtów danych			
5	Bajt danych nr 1	0..255	Młodszy bajt danych
6	Bajt danych nr 2	0..255	Starszy bajt danych
Przesyłanie czterech bajtów danych			
5	Bajt danych nr 1	0..255	Młodszy bajt danych
6	Bajt danych nr 2	0..255	
7	Bajt danych nr 3	0..255	
8	Bajt danych nr 4	0..255	Starszy bajt danych

Tablica 11
Programy, nastawy, dane: odczyt – struktura telegramów

Bajt danych	Opis	Wartość	Uwagi
1	Typ telegramu	16	Odczyt danych
2	Nr urządzenia żądającego dane	0..239	
3	Adres względny	0..255	Przesunięcie od adresu bazowego
Przesyłanie jednego bajta danych			
4	Bajt danych nr 1	0..255	
Przesyłanie dwóch bajtów danych			
4	Bajt danych nr 1	0..255	Młodszy bajt danych
5	Bajt danych nr 2	0..255	Starszy bajt danych
Przesyłanie czterech bajtów danych			
4	Bajt danych nr 1	0..255	Młodszy bajt danych
5	Bajt danych nr 2	0..255	
6	Bajt danych nr 3	0..255	
7	Bajt danych nr 4	0..255	Starszy bajt danych

Synchronizacja czasu

Jedno z urządzeń przyłączonych do magistrali CAN-Bus / RS485 może nadawać telegramy zawierające aktualny czas astronomiczny. Jest to potrzebne do synchronizacji pracy zegarów w poszczególnych urządzeniach. Strukturę telegramów synchronizacji czasu pokazano w tablicy 12.

Tablica 12
Synchronizacja czasu – struktura telegramów

Bajt danych	Opis	Wartość	Uwagi
1	Typ telegramu	17	Synchronizacja czasu
2	Rok	0..99	Ostatnie dwie cyfry roku
3	Miesiąc	1..12	
4	Dzień miesiąca	1..31	
5	Godzina	0..23	
6	Minuta	0..59	
7	Sekunda	0..59	
8	Dziesiątki milisekund	0..99	

Zapis / odczyt rejestrów

Urządzenie pracujące w sieci CAN-Bus / RS485 może udostępniać do zapisu / odczytu pewne rejestry ze swojej przestrzeni adresowej. Opisane telegramy umożliwiają zapis i / lub odczyt takich rejestrów. Dla każdego udostępnionego przez urządzenie rejestru powinny być w dokumentacji zdefiniowane następujące dane:

- adres rejestru z zakresu wartości 0x0000 do 0xFFFF,
- typ rejestru: jedno-, dwu- lub czterobajtowy,
- rodzaj dopuszczalnych operacji: czy jest dopuszczalny tylko odczyt, czy także zapis,
- sposób interpretacji wartości w rejestrze,
- w przypadku zapisywania do rejestru zakres dopuszczalnych wpisywanych wartości.

W zależności od typu rejestru wykorzystywane są odpowiednie telegramy. Na przykład do zapisywania / odczytywania rejestrów typu CHAR (1 bajtowe) są wykorzystywane telegramy typu 19, 20 i 21, dla rejestrów typu INT (2 bajtowe) – telegramy typu 22, 23 i 24, a dla rejestrów typu LONG (4 bajtowe) – telegramy typu 25, 26 i 27.

W dalszej części artykułu przedstawiono procedury zapisu i odczytu rejestrów typu INT. W przypadku rejestrów innych typów (CHAR i LONG) procedury są analogiczne, wykorzystywane są jedynie inne typy telegramów. Struktury telegramów zapisu, odczytu i potwierdzenia zapisu dla rejestrów typu CHART, INT, LONG podano w tablicach 13–21.

Zapis rejestru typu INT

1. Urządzenie wysyłające nową wartość rejestru wysyła telegram typu 22 o następującej zawartości:
 - bajt pierwszy – typ telegramu – wartość 22,
 - bajt drugi – odbiorca logiczny – wartość określająca numer urządzenia w sieci CAN, w którym zmieniana jest wartość rejestru,
 - bajt trzeci – młodszy bajt adresu rejestru,
 - bajt czwarty – starszy bajt adresu rejestru,
 - bajt piąty – młodszy bajt nowej wartości rejestru,
 - bajt szósty – starszy bajt nowej wartości rejestru.
2. Urządzenie, do którego jest adresowany powyższy telegram, po jego odebraniu zmienia wartość rejestru określonego adresem w telegramie na wartość określoną w telegramie, a następnie odsyła potwierdzenie zapisu nowej wartości rejestru jako telegram typu 24 o następującej zawartości:
 - bajt pierwszy – typ telegramu – wartość 24,
 - bajt drugi – nadawca logiczny – własny numer sterownika w sieci CAN,
 - bajt trzeci – młodszy bajt adresu rejestru,
 - bajt czwarty – starszy bajt adresu rejestru,
 - bajt piąty – młodszy bajt aktualnej wartości rejestru,
 - bajt szósty – starszy bajt aktualnej wartości rejestru.

Odczyt rejestru typu INT

1. Urządzenie odczytujące rejestr typu INT wysyła telegram typu 23 o następującej zawartości:
 - bajt pierwszy – typ telegramu – wartość 23,
 - bajt drugi – odbiorca logiczny – wartość określająca numer urządzenia w sieci CAN, w którym odczytujemy rejestr,
 - bajt trzeci – młodszy bajt adresu rejestru,
 - bajt czwarty – starszy bajt adresu rejestru.
2. Urządzenie, do którego jest adresowany telegram, po jego odebraniu odsyła telegram z aktualną wartością żądanego rejestru jako telegram typu 24 o następującej zawartości:
 - bajt pierwszy – typ telegramu – wartość 24,
 - bajt drugi – nadawca logiczny – własny numer sterownika w sieci CAN,
 - bajt trzeci – młodszy bajt adresu rejestru,
 - bajt czwarty – starszy bajt adresu rejestru,
 - bajt piąty – młodszy bajt aktualnej wartości rejestru,
 - bajt szósty – starszy bajt aktualnej wartości rejestru.

Tablica 13

Zapis rejestru CHAR

Bajt danych	Opis	Wartość	Uwagi
1	Typ telegramu	19	Zapis rejestru CHAR
2	Odbiorca logiczny	1..239	Według tablicy 2
3	Adres rejestru (bajt młodszy)	0..65535	Według DTR urządzenia
4	Adres rejestru (bajt starszy)		
5	Nowa wartość rejestru	0..255	Według DTR urządzenia

Tablica 17

Polecenie odczytu rejestru INT

Bajt danych	Opis	Wartość	Uwagi
1	Typ telegramu	23	Polecenie odczytu rejestru INT
2	Odbiorca logiczny	1..239	Według tablicy 2
3	Adres rejestru (bajt młodszy)	0..65535	Według DTR urządzenia
4	Adres rejestru (bajt starszy)		

Tablica 14

Polecenie odczytu rejestru CHAR

Bajt danych	Opis	Wartość	Uwagi
1	Typ telegramu	20	Polecenie odczytu rejestru CHAR
2	Odbiorca logiczny	1..239	Według tablicy 2
3	Adres rejestru (bajt młodszy)	0..65535	Według DTR urządzenia
4	Adres rejestru (bajt starszy)		

Tablica 18

Potwierdzenie zapisu lub odczyt rejestru INT

Bajt danych	Opis	Wartość	Uwagi
1	Typ telegramu	24	Potwierdzenie zapisu lub odczyt rejestru INT
2	Nadawca logiczny	1..239	Według tablicy 2
3	Adres rejestru (bajt młodszy)	0..65535	Według DTR urządzenia
4	Adres rejestru (bajt starszy)		
5	Aktualna wartość rejestru (bajt młodszy)	0..65535	
6	Aktualna wartość rejestru (bajt starszy)		

Tablica 15

Potwierdzenie zapisu lub odczyt rejestru CHAR

Bajt danych	Opis	Wartość	Uwagi
1	Typ telegramu	21	Potwierdzenie zapisu lub odczyt rejestru CHAR
2	Nadawca logiczny	1..239	Według tabeli 2.
3	Adres rejestru (bajt młodszy)	0..65535	Według DTR urządzenia
4	Adres rejestru (bajt starszy)		
5	Aktualna wartość rejestru	0..255	

Tablica 19

Zapis rejestru LONG

Bajt danych	Opis	Wartość	Uwagi
1	Typ telegramu	25	Zapis rejestru LONG
2	Odbiorca logiczny	1..239	Według tablicy 2
3	Adres rejestru (bajt młodszy)	0..65535	Według DTR urządzenia
4	Adres rejestru (bajt starszy)		
5	Nowa wartość rejestru (bajt najmłodszy)	0..4294967295	Według DTR urządzenia
6	Nowa wartość rejestru		
7	Nowa wartość rejestru		
8	Nowa wartość rejestru (bajt najstarszy)		

Tablica 16

Zapis rejestru INT

Bajt danych	Opis	Wartość	Uwagi
1	Typ telegramu	22	Zapis rejestru INT
2	Odbiorca logiczny	1..239	Według tablicy 2
3	Adres rejestru (bajt młodszy)	0..65535	Według DTR urządzenia
4	Adres rejestru (bajt starszy)		
5	Nowa wartość rejestru (bajt młodszy)	0..65535	Według DTR urządzenia
6	Nowa wartość rejestru (bajt starszy)		

Tablica 20

Polecenie odczytu rejestru LONG

Bajt danych	Opis	Wartość	Uwagi
1	Typ telegramu	26	Polecenie odczytu rejestru LONG
2	Odbiorca logiczny	1..239	Według tablicy 2
3	Adres rejestru (bajt młodszy)	0..65535	Według DTR urządzenia
4	Adres rejestru (bajt starszy)		

Tablica 21

Potwierdzenie zapisu lub odczyt rejestru LONG

Bajt danych	Opis	Wartość	Uwagi
1	Typ telegramu	27	Potwierdzenie zapisu lub odczyt rejestru LONG
2	Nadawca logiczny	1..239	Według tablicy 2
3	Adres rejestru (bajt młodszy)	0..65535	Według DTR urzędnika
4	Adres rejestru (bajt starszy)		
5	Aktualna wartość rejestru (bajt najmłodszy)	0..4294967295	
6	Aktualna wartość rejestru		
7	Aktualna wartość rejestru		
8	Aktualna wartość rejestru (bajt najstarszy)		

Tablica 22

Stany wewnętrzne – struktura telegramów

Bajt danych	Opis	Wartość	Uwagi
1	Typ telegramu	32	Stany wewnętrzne
2	Odbiorca logiczny	1..239	1..239 – według tablicy 2
3	Numer stanu	0..255	Definiowany w dokumentacji urzędnika
4..8	Zadawany stan (1–5 bajtów)	0..255	Definiowany w dokumentacji urzędnika

5. Protokół przesyłania informacji dowolnymi sieciami na potrzeby pracy systemu uzależnień wyłączników szybkich UPK2

Wobec konieczności dostatecznie szybkiego wyłączenia zwarć przez uzależnione wyłączniki szybkie, wymagane jest jak najszybsze przesłanie informacji. Niezbędne jest także ciągłe nadzorowanie sprawności transmisji, a przerwy przekraczające ustalone limity są traktowane jako niesprawność toru transmisyjnego.

Urządzenia transmisyjne stosowane do tego celu, wykorzystują najniższą, czyli pierwszą warstwę modelu OSI⁶ (sygnały fizyczne) i protokół UPK. Gwarantuje to, że bajt nadany przez nadawcę jest jednocześnie odebrany przez odbiorcę. Transmisja ta jest powszechnie wykonywana na sieci PKP za pomocą kabli telekomunikacyjnych TKD biegnących wzdłuż linii kolejowej. Na modernizowanych liniach kolejowych drogie kable TKD (Telekomunikacyjne Kable Dalekosiężne) są zastępowane kablami światłowodowymi. Stosowane również kable TKM (Telekomunikacyjne Kable Miejscowe) mają znacznie mniejsze przekroje par transmisyjnych niż kable TKD i przez to nie pozwalają na tworzenie rozległej sieci transmisyjnej.

Z tych powodów staje się konieczne stosowanie na potrzeby uzależnień wyłączników szybkich nowych sieciowych rozwiązań transmisyjnych. Rozwiązania takie mogą w pewnych sytuacjach wnosić opóźnienia niedopuszczalne w prawidłowej pracy systemu uzależnień. Zagrożeniem nie jest sama możliwość wystąpienia takich opóźnień, lecz brak możliwości ich wykrywania przy stosowaniu aktualnego protokołu UPK. Z tych powodów powstała konieczność opracowania nowego protokołu UPK2, który eliminuje wymienione zagrożenia.

Protokół UPK2 wykorzystuje rekordy protokołu UPK, które są pakowane w nowe ramki zawierające grupę dodatkowych informacji umożliwiających realizację nowych funkcji. Rozwiązanie takie pozwala na prostą konwersję telegramów uzależnieniowych pomiędzy protokołami UPK i UPK2. Konwersja sprowadza się do zapakowania rekordu UPK w ramkę UPK2 lub wyjęcie z ramki UPK2 części stanowiącej ramkę UPK.

Telegramy użytkownika

Kategoria „telegramy użytkownika” służy do przesyłania telegramów zdefiniowanych przez producentów urządzeń. Są to telegramy o numerach typu od 32 do 63. **W celu wykorzystania zdefiniowanej przez siebie struktury telegramu, producent powinien uzyskać akceptację jednostki aktualnie zarządzającej protokołem.** Wówczas zostaje przydzielony określony numer typu telegramu, a struktura telegramu jest opisana w dokumentacji protokołu.

Stany wewnętrzne – typ telegramu użytkownika zdefiniowany przez firmę „NEEL” – numer typu 32

Telegramy typu „stany wewnętrzne” służą do cyklicznego przesyłania do urządzenia jego stanów wewnętrznych. Może to być wykorzystane, np. do:

- przekazywania tekstu na wyświetlacz urządzenia,
- przekazywania stanu różnego rodzaju urządzeń sygnalizacyjnych lub alarmowych (np. lampki sygnalizacyjne, syrena alarmowa itp.),
- przekazywania zmiennych w czasie parametrów niezbędnych do pracy urządzenia,
- wymuszania innych działań, dla których stosowanie procedury poleceń z potwierdzeniem nie jest niezbędne ze względów bezpieczeństwa.

Niedopuszczalne jest stosowanie tych telegramów do wykonywania operacji sterowniczych urządzeniami. Strukturę telegramu typu „stanu wewnętrznego” przedstawiono w tablicy 22.

⁶ Model OSI – pełna nazwa ISO OSI RM, ang. *ISO Open Systems Inter-connection Reference Model* – model odniesienia łączenia systemów otwartych – standard zdefiniowany przez ISO oraz ITU-T opisujący strukturę komunikacji sieciowej. W tym przypadku dotyczy pierwszej warstwy zwanej też sygnałem fizycznym np. +12, -12 V [pl.Wikipedia.org/wili/model_OSI].

Protokół wymiany danych na potrzeby uzależnień wyłączników szybkich UPK2

Protokół służy do przesyłania rekordów danych protokołu UPK za pośrednictwem wszelkiego rodzaju sieci wykorzystujących wszelkie technologie różnych warstw modelu ISO. Transmisja na potrzeby uzależnień odbywa się między urządzeniami systemu uzależnień na odcinkach podstacja kabina-sekcyjna lub podstacja – podstacja.

Dane są przesyłane datagramami⁷. Każdy datagram zaczyna się jednym bajtem typu datagramu i dwoma bajtami długości datagramu (lo/hi). Typ określa strukturę dalszej porcji danych. Protokół przewidziany jest jako rozwojowy – w tym celu, po wyczerpaniu prawie wszystkich typów jednobajtowych, zaczęły być stosowane podtypy wyróżniane niektórymi z pozostałych wolnych typów. Wszystkie datagramy będą zabezpieczone CRC16 umieszczonym na końcu datagramu.

Struktura ramki

Ramka wysyłana w ramach protokołu UPK2 zawiera elementy, umieszczone w następującej kolejności:

STARTER	znacznik początku przesyłanego bloku danych (ramki),
TYP TELEGRAMU	pole informujące o budowie telegramu i jego funkcji,
ROZMIAR	liczba bajtów przesyłanego telegramu z CRC i Start, Stop,
ODBIORCA	numer odbiorcy telegramu,
NADAWCA	numer nadawcy telegramu,
CZAS NAD.	czas nadania telegramu,
NUMER TEL. NAD.	kolejny numer telegramu od nadawcy do odbiorcy,
NUMER TEL. ODB.	numer ostatniego telegramu odebranego przez nadawcę od odbiorcy,
CZAS ODB.	czas odebrania ostatniego telegramu odebranego przez nadawcę od odbiorcy,
TREŚĆ UPK	treść telegramu zgodna z protokołem UPK,
CRC	kontrola poprawności danych,
STOPER	znacznik końca ramki.

Obliczanie CRC

CRC oblicza się na podstawie wielomianu o potęgach $x^{16} + x^{15} + x^2 + x^0$.

Szczegółowy algorytm obliczania CRC wygląda następująco:

- na początku pakietu w rejestr 16-to bitowy wpisywane jest 0,
- każdy nadawany bajt jest xorowany z młodszymi 8-bitami rejestru, po czym rejestr jest przesuwany 8 razy w prawo, z wpisywaniem zer na bit lewy,
- za każdym razem, gdy z prawej strony będzie wysunięta jedyńka, rejestr jest xorowany z liczbą 101000000000001B (generator CRC 0xa001).

Do obliczenia CRC brana jest przyjmowana cała zawartość ramki z wyłączeniem pól STARTER, STOPER i CRC.

Znaki startera i stopera

W celu uniknięcia problemów czasowych z synchronizacją strumienia danych, datagramy będą opakowywane tak, że będzie wyróżniony jeden bajt startera F1h i jeden bajt stopera F2h. W opakowanym datagramie bajty te będą występowały tylko na jego początku i nigdzie indziej, jako nieinformacyjny starter i nieinformacyjny stoper. W pozostałej części datagramu:

- bajt F0h będzie zastąpiony sekwencją F0h, 00h,
- bajt F1h będzie zastąpiony sekwencją F0h, 01h,
- bajt F2h będzie zastąpiony sekwencją F0h, 02h.

W dalszej części artykułu przedstawiono listę dotychczas zdefiniowanych typów datagramów. W każdym typie mogą występować warianty zależne od znaczenia któregoś z kolejnych bajtów zawartości diagramu, więc lista może być wielopiętrowa. W kolejno opisywanych bajtach ramki z informacjami przyjęto następujące znaczenie:

- Odbiorca – urządzenie (miejsce), gdzie odbierany jest poniżej opisany telegram,
- Nadawca – urządzenie (miejsce), gdzie nadawany jest poniżej opisany telegram.

Proces wymiany informacji polega na współpracy par podstacja – kabina lub podstacja – podstacja. Oznacza to, że każdy nadawca jest jednocześnie odbiorcą informacji. Budowa ramki telegramu UPK2 ma następującą strukturę:

- bajt starter (F1h),
- bajt 1 – numer typu ramki (201–205),
- bajt 2 – długość datagramu lo,
- bajt 3 – długość datagramu hi,
- bajt 4 – nr odbiorcy lo,
- bajt 5 – nr odbiorcy hi,
- bajt 6 – nr nadawcy lo,
- bajt 7 – nr nadawcy hi,
- bajt 8 – rok lo,
- bajt 9 – rok hi,
- bajt 10 – miesiąc (1–12),
- bajt 11 – dzień (1–31),
- bajt 12 – godzina (0–23),
- bajt 13 – minuta (0–60),
- bajt 14 – sekunda (0–60),
- bajt 15 – milisekunda dziesiątki (0–100),
- bajt 16 – pojedyncze milisekundy (0–10),
- bajt 17 – nr kolejny wysłanej ramki do odbiorcy od nadawcy lo,
- bajt 18 – nr kolejny wysłanej ramki do odbiorcy od nadawcy hi,
- bajt 19 – nr ostatniej ramki przesłanej do nadawcy od odbiorcy lo,
- bajt 20 – nr ostatniej ramki przesłanej do nadawcy od odbiorcy hi,

⁷ Blok danych pakietowych przesyłany przez sieć komunikacyjną między komputerami lub abonentami sieci, zawierający wszelkie niezbędne informacje do przesłania danych z hosta źródłowego do hosta docelowego, bez konieczności wcześniejszej wymiany informacji przez te hosty. Podstawowa jednostka przesyłania danych w sieciach pakietowych, w których czas i kolejność dostarczenia poszczególnych jednostek danych nie są gwarantowane [pl.wikipedia.org/Wiki/datagram].

- bajt 21 – czas w dziesiątkach milisekund (0–255 czyli 0–2550 ms) jaki upłynął od odbioru ostatniej ramki nadanej przez nadawcę do odbiorcy.

Bajty zawierające treść zgodnie z protokołem UPK:

- ..
- ..
- bajt 22+UPK – crc16 lo,
- bajt 22+UPK – crc16 hi,
- bajt 22+UPK – stoper (F2h).

Wymagania techniczne

Do realizacji funkcji kontroli czasu przesyłania ramek konieczna jest synchronizacja czasu współpracujących ze sobą obiektów. Z tych powodów obiekty powinny mieć dostęp do czasu „satelitarnego” UTC. Ten czas powinien być używany w telegramach wysyłanych w ramach protokołu UPK2.

Dopuszcza się również awaryjną pracę bez dostępu do synchronizacji czasu UTC. W takim przypadku urządzenia kontrolują czas pełnego obiegu informacji nadawca – odbiorca – nadawca. Pomiar czasu takiego cyklu jest możliwy dzięki numeracji ramek i podawaniu czasu jaki upływa u odbiorcy od odebrania ostatniej ramki i jej numeru. Przy stosowaniu takiego mechanizmu konieczne jest zapamiętywanie czasów nadawania poszczególnych ramek.

Wyniki prób z proponowanymi zmodyfikowanymi protokołami

Wykonano podstawowe próby z protokołem UPK2. Dokonano sprawdzenia nowych funkcji, gdyż dotychczasowe wynikające z protokołu UPK zostały wystarczająco przetestowane w okresie blisko dwudziestoletniej eksploatacji. Nowe funkcje miały pozwolić na:

- kontrolę czasu przesyłania telegramów pomiędzy nadawcą a odbiorcą,
- kontrolę drożności i czasu transmisji odbiorca – nadawca – odbiorca,
- wykrywanie utraty telegramów.

W tym celu wykonano program testowy, generujący ramki protokołu UPK2 przesyłane przy wykorzystaniu protokołu TCP/IP lub UDP. Zawartość ramki w części uzupełnianej przez protokół UPK była wypełniana przypadkowymi znakami, ponieważ nie przewidywano potrzeby badania samego protokołu UPK.

Kontrola czasu przesyłania telegramów pomiędzy nadawcą a odbiorcą

Do wykonania tej funkcji jest wykorzystywany czas nadania telegramu przez nadawcę. U odbiorcy obliczana jest różnica pomiędzy zsynchronizowanymi czasami nadania i odbioru telegramu. Otrzymany czas powinien mieścić się w zadanym przedziale.

Kontrola drożności i czasu transmisji odbiorca – nadawca – odbiorca

Kontrola czasu pełnej pętli obiegu informacji od odbiorcy do nadawcy i od nadawcy do odbiorcy jest wykonywana

w sytuacji awaryjnej, gdy nadawca nie ma aktualnego zsynchronizowanego czasu UTC. Pomiar czasu tego obiegu jest możliwy dzięki numerowaniu ramek nadawanych i odsyłaniu numeru ostatnio odebranej ramki przez nadawcę. Warunkiem realizacji tej funkcji jest zapamiętywanie czasu nadania dla poszczególnych numerów ramek.

Wykrywanie utraty telegramów

Kontrola utraty odbieranych telegramów jest realizowana przez kontrolę ciągłości numerów telegramów otrzymywanych z danego kierunku.

6. Wnioski z realizowanej pracy

W pracy dokonano aktualizacji:

- przekazywania informacji na magistrali CAN-Bus/RS485 (PPM2) CNTK v3.0,
- protokołów na potrzeby uzależnień wyłączników UPK oraz opracowano i opisano protokół na potrzeby uzależnień wyłączników szybkich UPK2 v1.0.

Ponadto analiza wykazała, że:

1. Istnieje potrzeba przesyłania telegramów z magistrali CAN / PPM2 do systemów SCADA i odwrotnie. Zagadnienie to zostało samodzielnie rozwiązane przez firmy zajmujące się zdalnym sterowaniem. Polega na przetworzeniu telegramów w protokole PPM2 na telegramy zdalnego sterowania według własnych definicji, przesyłane sieciami WAN. Obecnie stosowane do tego celu standardy protokołów są opracowane przez poszczególne firmy i w związku z tym są ich własnością. Z tego powodu mogą występować problemy z udostępnianiem tych protokołów do powszechnego użytkowania. Celowym jest zapewnienie jednego ogólnodostępnego standardu w tej dziedzinie.
2. Obecny stan zabezpieczeń i urządzeń instalowanych w obiektach zasilania zaczyna stwarzać znaczne możliwości ich rozwoju funkcjonalnego. W szczególności dotyczy to rozwoju tzw. kanału inżynierskiego. Powstały również nowe normy międzynarodowe w zakresie interfejsów używanych w obiektach zasilania. Najkorzystniejszym i perspektywicznym rozwiązaniem w tym zakresie wydaje się stosowanie w obiektach światłowodowej sieci Ethernet. Ponieważ zastosowania trakcyjne mają swoją specyfikę, należałoby w najbliższych latach wprowadzić nowy standard przy współpracy z firmami, które mają doświadczenie w tej dziedzinie.

Bibliografia

1. Malicki J.: *Protokoły na potrzeby uzależnień wyłączników szybkich UPK1 i UPK2 v1.0, sierpień 2014 r.* (praca wykonana na zlecenie IK).
2. Malicki J.: *Protokół przekazywania informacji na magistrali CAN-Bus/RS485 - PPM2 v3.0,* (praca wykonana na zlecenie IK).