

# Magistrala CAN, część 3

## Zdecentralizowana wymiana danych

Dwie pierwsze części tego artykułu dotyczyły historii, normalizacji, podstawowej struktury i protokołu transmisji danych Sieci Obszaru Sterownika. W niniejszej, trzeciej części, położono nacisk na aspekty praktyczne.

### Detekcja błędów i ich korekcja

Jedną z najbardziej rzucających się w oczy cech magistrali CAN jest jej nadzwyczajna zdolność wykrywania wielu błędów podczas transmisji danych i odpowiedniego reagowania na nie. Ma ona odstęp Hamminga (nazywany także odstępem sygnałowym) równy 6. Odstęp sygnałowy między dwoma słowami dwójkowymi o tej samej długości jest liczbą bitów na odpowiadających sobie pozycjach, które mają różne wartości. Na przykład odstęp sygnałowy między słowami dwójkowymi 11011010 i 10000110 wynosi 4, ponieważ bity: 3-ci, 4-ty, 5-ty i 7-my (licząc od prawej) różnią się wartościami.

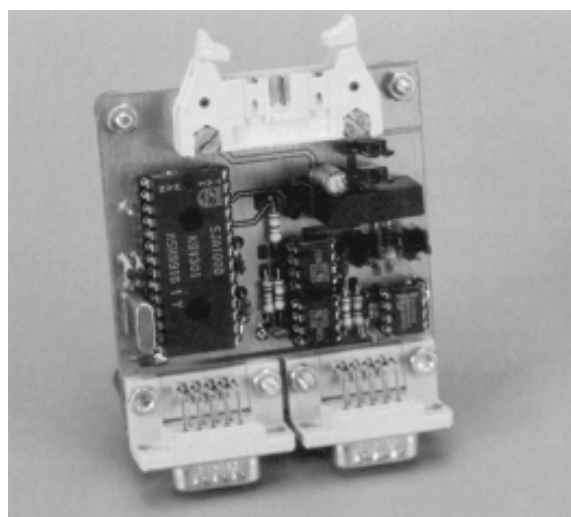
Magistralą CAN dane mogą być transmitowane z szybkością 500 kbitów/s. Na każde 0,7 s przypada jeden błędny bit spowodowany zewnętrznymi zakłóceniami. Sieć może pracować osiem godzin dziennie przez 365 dni w roku. Wbudowany sposób zabezpieczenia przed błędami gwarantuje, że przez 1000 lat pracy tylko jeden błąd nie będzie wykryty. Błędy mogą występować i oczywiście występują, ale skoro są rozpoznawane, to można je skorygować. Tylko nierozpoznane błędy mogą powodować, że fałszywe wyniki pomiarów będą przetwarzane.

### Wykrywanie błędów transmisji

W CAN zastosowano równocześnie kilka sposobów wykrywania błędów.

#### Detekcja błędnego bitu

Każda stacja zawsze odbiera zwrótnie swoją własną transmisję. Dlatego, jeżeli po fazie arbitrażu



jest tylko jedna stacja, która wysłała wiadomość do magistrali i odbiera zwrótnie inny, różniący się od wysłanego bit (stan magistrali), to jest oczywiste, że na magistrali wystąpił błąd. W takiej sytuacji stacja przełącza się na procedurę korekcji błędów. (patrz dalej).

#### Wykrywanie błędnych bitów dodatkowych

W specyfikacji CAN określa wyrażnie, że gdy w ramce danych jest transmitowanych kolejno więcej niż pięć bitów o tej samej wartości (na przykład siedem razy wartość zero w jakimś polu), to każda grupa pięciu bitów jest poprzedzana przez bit komplementarny (tutaj oczywiście 1). Ten wprowadzony bit, który oczywiście nie zawiera w ogóle żadnej informacji, jest nazywany bitem dodatkowym. Po zakończeniu odbioru, te bity są usuwane ze strumienia danych, tak że tylko pierwotna wiadomość jest przetwarzana.

Dodatkowe bity mogą być z łatwością użyte do kontroli błędów. Jeżeli odbiornik wykryje w ramce więcej niż pięć kolejnych bitów o tej samej wartości (lecz nie w polu EOF), to jest oczywiste, że to nie jest poprawny odczyt, i że

**Tab. 3. Porównanie parametrów systemów CAN 20A (format ramki standardowej) i CAN 20B (format ramki rozszerzonej)**

Parametr	CAN2.0A	CAN2.0B
Maksymalna liczba identyfikatorów	211	229
Liczba stacji (węzłów)	32	32
Szybkość transmisji [kbit/s]	5..125	5..1000
Liczba bajtów w ramce	0 - 8	0 - 8
Maksymalna długość ramki	117 bitów	13 bitów
Maksymalny zasięg sieci	patrz tekst	patrz tekst

Artykuł publikujemy na podstawie umowy z wydawcą miesięcznika "Elektor Electronics".

Editorial items appearing on pages 13..16 are the copyright property of (C) Segment B.V., the Netherlands, 1998 which reserves all rights.

wystąpił błąd podczas transmisji danych (wystąpił dodatkowy bit lub został „odwrócony” jeden lub więcej bitów). Wtedy odbiornik zawiesza działanie i uruchamia procedurę poprawiania błędów (zobacz dalej).

#### Detekcja błędu CRC

Proces ten polega, jak już wspomniano, na oszacowaniu sumy kontrolnej CRC w odbiorniku. Gdy sumy kontrolne: odebrana i obliczona różnią się, to odbiornik zmienia swoje działanie uruchamiając procedurę korekcji błędów (zobacz dalej).

#### Detekcja błędu potwierdzenia

W opisie formatu ramki (patrz rys. 6) wspomniano o bicie ACK (bit przerwy na potwierdzenie), który jest wysyłany przez stację jako bit recesywny. Wszystkie stacje, które poprawnie odebrały poprzednią ramkę nadpisują („przykrywają”) ten bit bitem dominującym. Stacja nadająca wykrywa to i „wie”, że przynajmniej jedna stacja odebrała jej dane poprawnie.

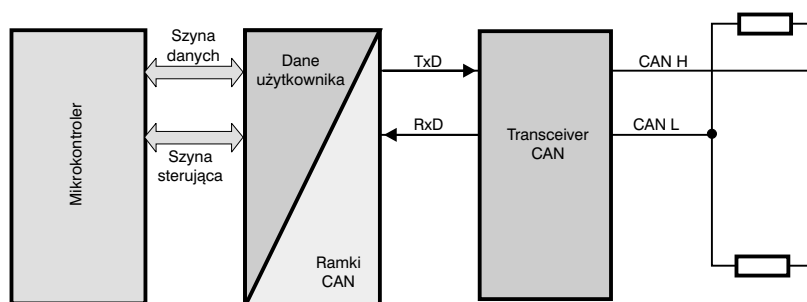
Jeżeli stacja nadająca stwierdzi, że jej bit w przerwie na potwierdzenie (ACK slot) nie został nadpisany, to „wie”, że żadna stacja nie odebrała jej wiadomości poprawnie. W takim razie, zawiesza swoje dotychczasowe działanie i wywołuje procedurę korekcji błędów (zobacz dalej).

#### Detekcja błędu formatu

W tym procesie wykorzystywany jest fakt, że w ramce jest kilka pól, które muszą zawsze mieć ustaloną zawartość: w polu ogranicznika CRC (bit końca CRC), w polu ogranicznika potwierdzenia i w polu EOF są zawsze bity recesywne. Jeżeli w tych polach zostanie wykryty bit dominujący, to taki stan może być tylko spowodowany przez błąd transmisji danych. Wówczas stacja nadająca uruchamia także procedurę korekcji błędów.

#### Korekcja błędów

Procedura korekcji błędów w wypadku błędu transmisji danych jest realizowana w dwóch wariantach.



Rys. 9 Schemat blokowy trójstopniowej Sieci Obszaru Sterownika (CAN).

Po pierwsze, ramki w których został stwierdzony jakiś błąd są natychmiast odrzucane przez odpowiednią stację i nie przetwarzane. Po drugie, jeżeli któraś ze stacji systemu wykryje jakiś błąd, to wysyła natychmiast ramkę informującą o błędzie, która składa się z sześciu dominujących bitów (sygnalizacja błędu) i ogranicznika ramki błędu zawierającego osiem bitów recesywnych. Wskutek tego wszystkie bity recesywne na magistrali są nadpisywane, tak że występuje na niej tylko sześć dominujących bitów. Jest to jednak naruszenie przyjętej zasady, że nie więcej niż pięć kolejnych bitów może mieć tę samą wartość.

Wszystkie pozostałe stacje dołączone do magistrali wykrywają ten stan i uznają dopiero co odebraną ramkę jako błędną (wadliwą), odrzucają ją i także wysyłają ramkę sygnalizującą o błędzie. Inaczej mówiąc, stacja która wykryła błąd celowo „uszkadza” całą transmitowaną ramkę, tak że wszystkie stacje dołączone do magistrali odbierają ją jako błędną. To oznacza, że o jakimś błędzie lokalnym w jednej stacji są natychmiast poinformowane wszystkie pozostałe stacje. Głównym założeniem sieci jest, żeby wszystkie stacje odbierały poprawne dane, które mogą być dalej przetwarzane lub żeby wszystkie stacje odbierały błędne dane, które będą odrzucane. Pierwotna stacja nadająca stwierdza oczywiście, że ramka którą wysłała jest z błędem, poprawia tą wiadomość i natychmiast wysyła ponownie.

#### Błąd wewnątrz stacji

Co się stanie, gdy stacja sama stwierdzi, że jest uszkodzona, wysyła dane z nieodpo-

wiednią szybkością lub jest stacją, która odbiera tylko błędne dane? Taka stacja mogłaby stale wysyłać ramkę sygnalizującą błąd i tym samym zablokować całą sieć. CAN jest odpowiednio zabezpieczony przed takim zdarzeniem, ale brak miejsca nie pozwala na opisanie tego w tym artykule.

#### Podsumowanie

Parametry dwóch wersji systemu CAN, tj. CAN 20A (format ramki standardowej) i CAN 20B (format ramki rozszerzonej) porównano w **tablicy 3**.

Chociaż CAN jest efektywnym i bardzo niezawodnym systemem do przesyłania danych, to Czytelnik i potencjalny użytkownik systemu może zapytać jak można praktycznie zastosować ten system? Są w nim bity dominujące i recesywne, 11-bitowy identyfikator, 15-bitowa suma kontrolna CRC, 1-bitowy ogranicznik, 7-bitowe pole EOF, 6-bitowa ramka błędu i wiele innych „dziwności”. Żadna z nich nie kojarzy się bezpośrednio ze strukturą danych mikrokontrolera 8- lub 16-bitowego.

Jak więc jest możliwe zaprogramowanie mikrokontrolera zgodnie z protokołem sieci? Jeżeli o to chodzi, to przyszły konstruktor nie powinien się tym martwić. Dla tej sieci jest bowiem dostępnych mnóstwo gotowych, niedrogich elementów składowych, modułów. Są one dostarczane przez producentów układów scalonych dla systemu CAN, co spowodowało, że ta sieć stała się tak popularna, w tak krótkim czasie.

Typowe interfejsy CAN składają się głównie, jak to widać na schemacie blokowym (**rys.**

9), z trzech chipów. Jedynym zadaniem mikrosterownika jest wpisywanie bajtów danych (0..8), które mają zostać wysłane do układu scalonego protokołu CAN, wypełnianie pola identyfikatora i pola DLC oraz odpowiednie ustawienie bitu RTR. Pozostałe elementy procesu przetwarzania:

- obliczanie sumy kontrolnej CRC,
- dodawanie pozostałych pól,
- łączenie się z magistralą,
- transmisja danych,
- wykrywanie i usuwanie błędów są wykonywane przez układ scalony sterownika CAN.

Dane są wprowadzane do magistrali za pośrednictwem układu scalonego transceivera CAN, który jest bezpośrednio połączony z magistralą. Mikrosterownik otrzymuje wtedy potwierdzenie pomyślnego wysłania danych, albo komunikat o błędzie, poczym podejmuje odpowiednie działanie.

Mniej więcej to samo dzieje się przy odbiorze danych. Sterownik CAN, za pośrednictwem układu scalonego transceivera CAN, otrzymuje ramki CAN z magistrali, ponownie sprawdza sumę kontrolną, usuwa z ramek wszystkie zbędne pola i do mikrosterownika przesyła otrzymane dane, albo komunikat o błędzie.

Czytelnik z pewnością już zauważył, że wymagania sprzętowe i programowe interfejsu magistrali CAN nie są wielkie. Mikrosterowniki, zawierające sterownik CAN, są już dostępne na rynku. Umożliwiają one skonstruowanie dwustopniowego interfejsu CAN.

Przed przystąpieniem do omówienia konstrukcji interfejsu przedstawionych jeszcze zostanie kilka dodatkowych informacji.

### Filtrowanie akceptacyjne

Z części 2 wiadomo, że CAN działająca w standardowym formacie ramek (CAN20A) jest w stanie przetwarzać do 2048 różnych identyfikatorów. Oczywiście nie jest niezbędne, aby każda ze stacji przyłączonych do magistrali otrzymywała wszystkie ramki danych. Na przykład może być tak, że dla stacji K istotne są tylko ramki z identyfikatorami 129, 1345 i 1999, a 2045 pozostałych nie żadnego znaczenia. Bardzo pożądaną staje się więc wprowadzenie takiej selekcji identyfikatorów, aby do mikrosterownika nie docierały ramki zbędne. Selekcja ta nazywa się filtracją akceptacyjną. Umożliwia ona takie zaprogramowanie sterownika CAN, żeby sprawdzał wszystkie otrzymywane ramki (wraz z korekcją błędów), ale do mikrosterownika wysyłał tylko ramki o określonych identyfikatorach. Bez konieczności dokonywania przez mikrosterownik zbytecznych porównań przetwarzanie staje się szybsze. Do filtrowania akceptacyjnego można użyć dwóch różnych układów scalonych.

### Układ BasicCAN

Układ ten zawiera prosty filtr o szerokości ośmiu bitów, pozwalający na jedynie zgrubną selekcję wstępną. Polega ona na tylko grupowym przepuszczaniu identyfikatorów, na przykład

700..707. Wybór pojedynczego identyfikatora jest wtedy możliwy tylko po przeprowadzeniu dalszej selekcji z użyciem mikrosterownika. Przeznaczone dla danej stacji ramki zdalne również przechodzą przez filtr przed dotarciem do mikrosterownika. Tylko wtedy może on wygenerować w odpowiedzi właściwe dane i skierować je do sterownika CAN.

### Układ FullCAN

Układ ten umożliwia dokładne zaprogramowanie i selekcję pojedynczego identyfikatora. Innymi słowy, układ może zostać przystosowany do akceptowania jednej lub określonej liczby ramek, na przykład tylko ramki z identyfikatorem 798. Jednak układ ten nie będzie przepuszczał dużej liczby ramek o różnych identyfikatorach, ponieważ program sterownika jest ustalony.

Jeżeli więc ma być odbierane wiele ramek o różnych identyfikatorach, lepiej wybrać chip BasicCAN. Trzeba jednak pamiętać, że wtedy znaczna część procesu selekcji z konieczności przejmie mikrosterownik, którego moc przetwarzania będzie musiała być zwiększona.

Zaletą układu FullCAN jest możliwość programowania przez mikrosterownik w układzie scalonym sterownika CAN odpowiedzi na zdalną ramkę. Gdy taki układ otrzymuje dozwoloną zdalną ramkę dla odnośnej stacji, może wysłać w odpowiedzi ramkę danych bez interwencji mikrosterownika.

Wraz z nieuniknionym rozwojem technologii różnice pomiędzy układami BasicCAN i FullCAN stają się coraz mniej wyraźne. Chipy FullCAN stają się także coraz sprawniejsze, mogą wybierać coraz większe liczby identyfikatorów i przechowywać coraz więcej rejestrów danych. Najnowsze układy sterowników CAN mogą za pomocą programu przełączać się pomiędzy dwoma trybami działania.

### Zgodność pomiędzy 20A i 20B

Jak już wiadomo z dyskusji o formatach ramek, istnieje *Format Standardowy* z 11-bitowymi identyfikatorami i *Format Rozszerzony*

**Tab. 4. Podstawowe parametry scalonego interfejsu SJA1000.**

Układ scalony sterownika CAN	SJA1000 (Philips Semiconductors)
Interfejs mikrosterownika	Może zostać dostosowany do mikrosterowników Intel i z nim zgodnych lub do mikrosterowników Motorola i z nim zgodnych.
Tryb działania 1	Rozmieszczeniem wyprowadzeń, sprzętowo i programowo zgodny z PCA82C200, CAN20A i bierny CAN20B. Standardowy format ramek. Szybkość transmisji danych do 1Mb/s. Filtr akceptacyjny BasicCAN.
Tryb działania 2	Standardowy i rozszerzony format ramek. Szybkość transmisji danych do 1 Mb/s. Możliwość CAN 20B. Rozszerzony filtr akceptacyjny o własnościach BasicCAN.
Układ scalony transceivera	Szybki CAN, zgodny z ISO/DIS11898. Szybkość transmisji do 1 Mb/s. Zabezpieczenie wewnętrzne przed zakłóceniami wytwarzanymi przez pojazdy silnikowe. Zabezpieczenie wewnętrzne przed zwarciem i przegrzaniem. Nie zasilone węzły (stacje) nie oddziałują na magistralę. Pozwala konstruować CAN o liczbie węzłów do 110.

z 29-bitowymi identyfikatorami. Dlatego przy wyborze sterownika CAN, gdy w magistrali są używane oba formaty (co jest zupełnie możliwe i dopuszczalne), należy zachować dużą ostrożność.

#### *Sterowniki z możliwym 20A*

Te sterowniki mogą przetwarzać tylko ramki standardowe, a po odebraniu ramki rozszerzonej generują komunikat o błędzie. Może to całkowicie wstrzymać działanie systemu, więc sterowniki tego rodzaju mogą być używane tylko systemach z ramkami wyłącznie standardowymi.

#### *Sterowniki z możliwym 20A i biernymi cechami 20B*

Układy te akceptują ramki rozszerzone z 29-bitowymi identyfikatorami, przeprowadzają próbę błędu i odpowiadają bitem ACK (potwierdzenia) albo ramką błędu.

Łączność nie zostaje zakłócona, ale rozszerzone ramki danych nie są ani zapisywane ani przepuszczane, ponieważ układy te są przewidziane jedynie do przetwarzania ramek formatu standardowego. Niemniej w pełni nadają się do użytku w systemach hybrydowych.

#### *Sterowniki z możliwym 20B*

Te sterowniki przetwarzają, przechowują i przepuszczają ramki zarówno formatu standardowego jak i rozszerzonego.

Przy podejmowaniu decyzji o zakupie sterownika CAN, czy mikrosterownika zawierającego sterownik CAN, napotyka się na tak duży ich wybór, że opłaca się przejrzeć przedtem internetowe witryny najbardziej znanych producentów tych układów, Hitachi, Intel, Motorola, NSC, Philips, SGS, Siemens, Temic i Texas Instruments.

**EE**