

Magistrala CAN, część 1

Zdecentralizowana wymiana danych

Rozprzestrzenienie się sieci komunikacji lokalnej w systemach przemysłowych wydaje się nieodwołalne. Coraz powszechniej stosowane są magistrale CAN, Profibus, LON, ASI, Interbus-S, FIP, EIB, eBus i wiele innych.

Czas pokaże, kiedy te ustabilizowane już technologie z protokołami zaimplementowanymi w małych chipach, niskimi cenami i łatwą eksploatacją staną się dostępne dla mniejszych firm inżynierskich.

W trzech częściach artykułu przedstawimy technologię przesyłania informacji, wykorzystującą system magistrali CAN.

Interfejs będzie opisany w sposób prosty i praktyczny.

Całość zakończy opis konstrukcji i oprogramowania sieci z magistralą CAN.

W pierwszej części artykułu omówimy pokrótce historię i standaryzację magistrali CAN (Controller Area Network). Znajdzie się tu również charakterystyka warstwy fizycznej.

W drugiej części zostanie objaśniona warstwa łącza. Część ta pomieści również, w formie tabelarycznej, elementy składowe budowy CAN różnych producentów. Dalej wstępny opis uniwersalnego interfejsu magistrali CAN, który może być wykorzystany do wykonania systemu z mikrokontrolerem lub mikroprocesorem, odpowiedniego dla zastosowania w sieci magistrali CAN.

W części trzeciej opiszemy konstrukcję, oprogramowanie i użytkowanie małej sieci magistrali CAN w połączeniu z komputerem PC i z kartą mikrokontrolera.

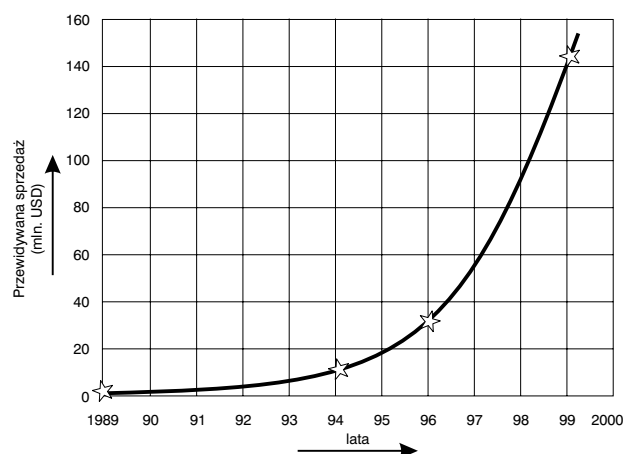
Opracowanie magistrali CAN

W początku lat dziewięćdziesiątych, międzynarodowy przemysł samochodowy stanął przed dwoma problemami dotyczącymi rozwoju samochodów prywatnych i pojazdów dostawczych.

Pierwszy odnosił się do postulatów poprawienia komfortu pojazdów: elektrycznie podnoszonych szyb, regulacji siedzeń i lusterek, podgrzewanych siedzeń, elektronicznego sterowania klimatyzacją, jak również wyposażenia audiowizualnego i satelitarnie sterowanych systemów nawigacyjnych (GPS - Global Positioning System).

Drugim i ważniejszym był problem bezpieczeństwa pojazdów, nie tylko z indywidualnego punktu widzenia, ale również dla spełnienia coraz surowszych międzynarodowych przepisów odnośnie bezpieczeństwa: centralnego zamka drzwi, systemów antykradzieżowych, ABS (systemów przeciwdziałających poślizgowi kół), jak również ekonomicznego i przyjaznego dla środowiska sterowania pracą silnika.

Obu problemom stawiono czoła poprzez intensywną elektronizację komunikacji wewnątrz pojazdu zawierającego wiele urządzeń. Oszacowano, że pojazdy wyprodukowane około roku 2005 będą zawierać do 100 mikrokontrolerów i wszystkie powinny mieć możliwość komunikowania się ze sobą. Jest oczywiste, że rezultatem wszystkich tych ścieżek komunikacji będzie jeszcze większa i bardziej rozgałęziona płatanina kabli. Na przykład, w nowoczesnym, dobrej jakości samochodzie silnik waży blisko 100kg (220 funtów) i jest do 2000 metrów kabli. Co więcej, u typowego wielkiego producenta samochodów może być stoso-



Rys. 1. Oszacowanie światowej sprzedaży układów scalonych CAN.

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

Editorial items appearing on pages 21..24 are the copyright property of (C) Segment B.V., the Netherlands, 1998 which reserves all rights.

| Model warstwowy ISO/OSI | Warstwa magistrali CAN | | |
|--|--|------------------------|--------------------------------|
| Warstwa 8 | | | |
| Aplikacja: "Urządzenie na magistrali" | CANopen | DeviceNet | Smart Distributed System (SDS) |
| Warstwa 7 "Warstwa aplikacji" | CAL: CAN Warstwa aplikacji dla aplikacji przemysłowych | Specyfikacje DeviceNet | Specyfikacje SDS |
| Warstwy 3..6 | Puste!!! | | |
| Warstwa 2 "Warstwa łącza danych" | LLC: Logical Link Control MAC: Medium Access Control zgodnie z ISO 11898 Rezultat: Specyfikacje CAN 2.0A, CAN 2.0B | | |
| Warstwa 1 "Warstwa fizyczna" | "Low-Speed CAN" ISO 11519-2 | | "High-Speed CAN" ISO 11898 |

wanych do 600 różnych typów wiązek kabli.

Ponieważ jest to sytuacja dłużej nie do utrzymania, przemysł samochodowy zaczął się rozglądać za nowymi sposobami komunikacji i znalazł je w przemyśle komputerowym. Oczywiście, system magistrali stosowanych w tym przemyśle, wymagał adaptacji dla zastosowania w pojazdach, przede wszystkim z następujących powodów:

- przenoszenia danych z małą i dużą szybkością w zakresie od 5kb/s do 1Mb/s,
- bezbłędnego przenoszenia danych,
- optymalnego przenoszenia mikrostrumieni danych, takich jak uzyskiwane z czujników lub urządzeń wykonawczych, to jest złożonych z 0..8 bajtów na komunikat,

- łatwości utrzymania,
- niskich kosztów w masowej produkcji,
- prostoty konstrukcji magistrali (media magistrali, topologia magistrali) dla łatwej integracji w pojeździe.

Niestety, wielcy producenci samochodów opracowali już swoje własne magistrale, które nie są kompatybilne z odpowiednikami u innych producentów. Wszyscy próbowali doprowadzić do przyjęcia swoich systemów jako systemu międzynarodowego, to jest do zaakceptowania ich jako międzynarodowego standardu i uzyskania dla siebie wszystkich oczywistych korzyści ekonomicznych i komercyjnych.

Nie wszystkie systemy mogły być łatwo zestandaryzowane. Zasadniczo, przyjęły się cztery z nich: CAN (w wersjach o małej i dużej szybkości), VAN, J1850CP i J1850DLC. VAN (zestandaryzowany) i pozostałe (niestandaryzowane) zostały zarzucone w połowie lat 90. na korzyść systemu CAN. Dziś system CAN jest światowym liderem na polu magistrali dla pojazdów. Jako taki jest obecnie stosowany nie tylko w samochodach marek luksusowych, jak

Mercedes, Lexus, Jaguar i Chrysler, ale również w tych mniej prestiżowych, jak Fiat i Volkswagen.

Nie tylko przemysł samochodowy odkrył zalety magistral, ale również przemysły automatyzacji i przetwórczy. Przemysły te wykorzystują pomysł CAN dla pomiarów, kontroli i sterowania w systemach SPS (Standard Positioning Service), robotach i silnikach. Pomysł ten jest również stosowany w budownictwie, w sterowaniu windami, systemami automatyzacji laboratoriów, systemami czujnikowo-wykonawczymi i innymi.

Protokół CAN jest dostępny w krzemowym chipie, tak że użytkownik nie musi już koncentrować się na drobniejszych szczegółach technologii komunikacyjnej. Układy CAN są po prostu zintegrowane jako inteligentne peryferyjne bloczki składowe w istniejących systemach mikrokontrolerowych lub tych, które dopiero zostaną zbudowane.

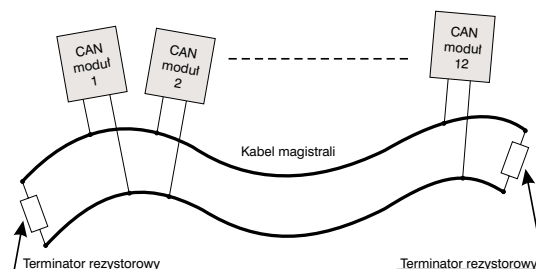
Wolne od kłopotów utrzymanie i stosowanie, jak również gwałtownie spadające ceny układów scalonych CAN czynią CAN magistralami cieszącymi się wielkim zainteresowaniem i wielce przydatnymi dla konstruktorów małych, zdecentralizowanych sieci komunikacyjnych.

Na rys. 1 przedstawiono oszacowanie wielkości światowej sprzedaży układów scalonych CAN. Koszty kompletnych sieci CAN, pochodzących od różnych producentów półprzewodników są podawane w USD.

Standaryzacja

Jeśli struktura systemu komunikacyjnego ma być powszechnie akceptowana, należy znaleźć odpowiedź na kilka pytań dotyczących standardu:

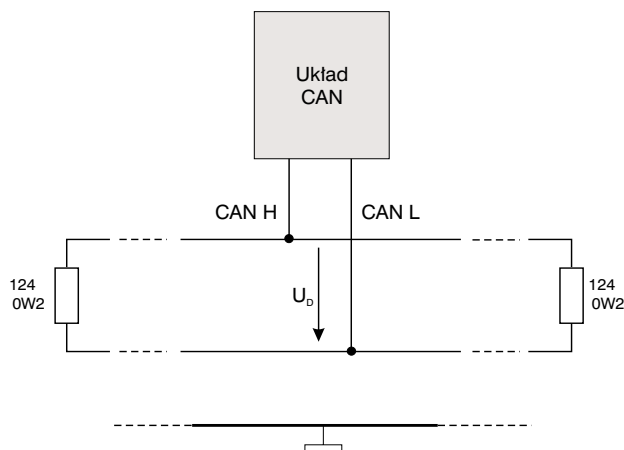
- Jak fizycznie (elektrycznie, logicznie) będą zorganizowane różne części sieci?



Rys. 2. Topologia magistrali CAN.

Tab. 1. Poziomy bezwzględne linii magistrali w odniesieniu do (lokalnej) masy zgodnie z ISO11898.

| Napięcie na magistrali | Stan magistrali | |
|--|------------------------|---------------------------|
| | recesywny {ustępujący} | dominujący {przeważający} |
| CANH | 2,5V | 3,5V |
| CANL | 2,5V | 1,5V |
| dopuszczalne napięcie różnicowe $U_0 = \text{CANH} - \text{CANL}$ | 0 - 0,5V | 0,9 - 2,0V |



Rys. 3. Dołączanie elementu (stacji) do magistrali CAN.

- Jak będzie wyglądać wynikowa topologia sieci?
- Jak dane będą porządkowane i przesyłane poprzez odpowiednie medium (kabel czy światłowód lub powietrze za pomocą podczerwieni)?
- Jakie są reguły wymiany danych pomiędzy różnymi częściami?
- Jak zapobiegać, rozpoznawać, korygować błędy przesyłania danych?
- Jak jest sformowany protokół przesyłania danych?
- Jak jest zorganizowany dostęp do medium przesyłania danych dla wszystkich elementów (stacji) związanych z ich przesyłaniem?
- Jak są rozstrzygane konflikty, gdy kilka elementów zechce przesyłać dane w tym samym czasie? (dotyczy to dostępu do medium, czyli tak zwanego arbitrażu).

Dopóki dotyczy to odbioru danych, nie ma tu wielu problemów, ponieważ z reguły do medium może być dołączonych wiele odbiorników, z których wszystkie mogą w tym samym czasie bez żadnych trudności odbierać dane. Ogólnie, w dowolnym systemie komunikacyjnym w danym momencie aktywny powinien być tylko jeden nadajnik, podczas gdy odbiorników może być kilka.

Odpowiedzi na powyższe pytania muszą być jednoznaczne, jeśli system komunikacyjny ma być stosowany w sposób sensowny i akceptowalny w całym Świecie.

W początkach lat dziewięćdziesiątych International Standard Organisation (ISO) zaczęła układać

międzynarodową normę dla magistral pojazdów, kiedy to magistrala CAN znacząco ugruntowała swoją mocną pozycję.

Podstawą procesu standaryzacji komunikacji danych, jest siedmiowarstwowy model odniesienia ISO/OSI. W przypadku niektórych syste-

mów komunikacyjnych, włącznie z systemem magistrali dla pojazdów, warstwy ISO 3..6 są puste, tak że dla magistrali CAN tylko warstwy 1, 2 i 7 są wyspecyfikowane szczegółowo.

Warstwa 1 - warstwa fizyczna

W tej warstwie znajdują się specyfikacje medium transmisji danych, złączy, poziomów przesyłania oraz elementów nadawczych i odbiorczych. Dwoma standardami związanymi z CAN są:

ISO11529-2: CAN o małej szybkości. Jej podstawą jest opracowanie zapoczątkowane przez firmę Bosch we wczesnych latach osiemdziesiątych i kontynuowane, przy silnym wsparciu Intel'a, aż do zintegrowania protokołu w układzie scalonym. Mała szybkość odnosi się do szybkości przesyłania od 5kb/s do 125kb/s.

ISO11898: CAN o dużej szybkości. Ten standard dotyczy szybkości przesyłania danych do 1Mb/s.

Warstwa 2 - warstwa łącza danych

Warstwa ta określa jak staje się dostępne medium przesyłania danych, gdy jakaś część systemu chce wysłać dane, jak jest tworzony komunikat (adres, sterowa-

nie, dane i zabezpieczenie przed błędami) i jaki jest protokół przesyłania danych. Normy te można również znaleźć w ISO11898.

Ponadto, specyfikacja CAN 1991 w Warstwie 2 była modyfikowana, tak że dziś są tam dwie jej wersje: CAN2.0A i CAN2.0B. Do podobieństw i różnic pomiędzy tymi dwiema wersjami powrócimy w drugiej części artykułu.

Całe lata trwało opracowywanie trzech obszernych gałęzi CAN dla różnych aplikacji: CANopen, DeviceNet i Smart Distributed System (SDS). Ponieważ specyfikacje te są naprawdę obszerne, nie będziemy ich śledzić w tym artykule. Powiemy tylko tyle, że są one kompatybilne z Warstwami 1 i 2.

Szczegółowe informacje odnośnie CANopen, DeviceNet i SDS można znaleźć w Internecie: <http://www.can-cia.de>.

Charakterystyka

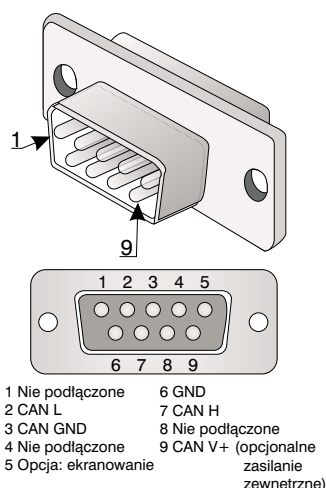
W warstwie fizycznej jest zawarta specyfikacja topologii sieciowej magistrali CAN i dołączania elementów (stacji) do medium magistrali.

Termin topologia sieci obejmuje fizyczną konstrukcję systemu komunikacyjnego i daje tym samym odpowiedź na pytanie: „jak elementy (stacje) są połączone z medium przesyłania danych?”

CAN wykorzystuje tak zwaną topologię magistrali, to oznacza, że wszystkie elementy są połączone z pojedynczą skrętką pary przewodów (ekranowaną lub nie), zakończoną na obydwu końcach odpowiednimi impedancjami (patrz rys. 2). Taka organizacja zapewnia, że każda stacja może komunikować się z każdą w sieci bez żadnych ograniczeń.

Tab. 2. Współzależność pomiędzy szybkością przesyłania danych, długością magistrali, medium magistrali i impedancją zamykającą magistrali.

| Długość magistrali | Kabel magistrali | | Rezystancja zamykająca magistrali | Maksymalna szybkość przesyłania danych |
|--------------------|------------------|--|-----------------------------------|--|
| | rezystancja | powierzchnia przekroju poprzecznego kabla | | |
| 0 - 40m | 70mΩ/m | 0,25 - 0,34mm ² AWG23, AWG22 | 124Ω (1%) | 1Mb/s przy 40m |
| 40 - 300m | <60mΩ/m | 0,34 - 0,5mm ² AWG22, AWG20 | 127Ω (1%) | 500kb/s przy 100m |
| 300 - 600m | <40mΩ/m | 0,5 - 0,6mm ² AWG20 | 150Ω do 300Ω | 100kb/s przy 500m |
| 600m - 1km | <26mΩ/m | 0,75 - 0,8mm ² AWG18 | 150Ω do 300Ω | 50kb/s przy 1km |



Rys. 4. Rozkład wyprowadzeń złącza magistrali CAN.

Układ nadawania/odbioru sieci CAN jest połączony z medium magistrali poprzez dwa doprowadzenia: CAN High (CANH) i CAN Low (CANL) (patrz rys. 3). Ze względu na wymagane zabezpieczenie przed błędami, do rzeczywistego przesyłania danych stosuje się różnicowe sygnały napięciowe. Oznacza to, że różnica napięcia pomiędzy obydwoma liniami magistrali jest skwantowana.

Standard ISO11898 specyfikuje dwa różne zakresy napięcia różnicowego dla reprezentacji danych: recesywny i dominujący. Istnieje ważna przyczyna, że zwykła logika poziomów 0 i 1 tu nie jest stosowana i do tego wrócimy. Na razie zauważmy, że:

- jeśli napięcie różnicowe pomiędzy CANH i CANL 0,5V, status linii jest recesywny,
- jeśli napięcie różnicowe 0,9V, status jest dominujący.

Poziom nominalny linii magistrali, to jest poziom poszczególnych linii w odniesieniu do masy lokalnej, przedstawiono w tab. 1.

W praktyce poziomy te mają oczywiście jakieś tolerancje, tak że napięcie różnicowe może osiągać poziom maksymalny dopuszczalny, podany w ostatnim wierszu tabeli 1.

Specyfikacja (CANL) w standardzie ISO11519-2 jest nieco odmienna, ale ponieważ standard ISO11898 może być stosowany zarówno dla dużych jak i dla małych szybkości, obecnie jest stosowana ta specyfikacja.

Użytkownicy nie muszą sami zajmować się konstrukcją łącza nadawania/odbioru, ponieważ u większości producentów są do tego celu dostępne gotowe układy scalone. Są one zoptymalizowane, szczególnie pod względem zakłóceń elektromagnetycznych (EMC), zajmowanej powierzchni płytki drukowanej i przeciążeń termicznych (w przypadku zwarcia CANH lub CANL) i wyjściowego standardu poziomów sygnałów CAN. Wszystkim, co jest niezbędne dla zestawienia łącza CAN, to dołączenie go do linii magistrali.

A wszystko, co należy zrobić, to upewnienie się, dla którego standardu CAN układ scalony został zbudowany: ISO11519-2 lub ISO11898. Preferowany powinien być ten drugi.

Należy zauważyć, że w praktyce wykorzystywane są również inne sposoby różnicowego przesyłania danych, które mogą posłużyć do przesyłania sygnałów CAN, na przykład RS485.

Ostatnie pytania, które należy postawić odnośnie systemu magistrali CAN, to:

- Jaka jest maksymalna długość magistrali dla danej szybkości przesyłania?
- Ile elementów (stacji) można dołączyć do magistrali?

Odpowiedź na te pytania zależy wyłącznie od zastosowanego medium magistrali. W tab. 2 przedstawiono korelację pomiędzy szybkością przesyłania danych, długością magistrali, medium magistrali i impedancją zakończenia magistrali.

Najlepsze medium przesyłania danych to skrętka pary przewodów o powierzchni przekroju poprzecznego 0,34..0,6mm², podczas gdy impedancja zakończenia magistrali powinna wynosić około 127Ω. Rezystywność kabla nie powinna być większa niż 60mΩ/m, który to warunek jest spełniony, gdy powierzchnia przekroju poprzecznego jest większa niż 0,30mm².

Gdy stacja nie jest dołączona bezpośrednio do magistrali CAN, to należy wziąć pod uwagę długość linii doprowadzeniowych. Linie te nie powinny być dłuższe niż 2 metry, jeśli szybkość przesyłania danych ma wynosić 250kb/s i nie dłuższe niż 30cm, jeśli szybkość przesyłania danych ma być większa. Całkowita długość wszystkich linii doprowadzeniowych nie powinna przekraczać 30 metrów.

Na koniec, uwaga dotycząca Warstwy 1. Wszystkie złącza i rozkład ich wyprowadzeń są zstandaryzowane.

EE

Magistrala CAN, część 2

Zdecentralizowana wymiana danych

W pierwszej części artykułu opisano historię, ustalenia normalizacyjne i podstawową strukturę systemu komunikacyjnego CAN opracowanego przez niemiecką firmę Roberta Boscha. W drugiej części skupimy się na protokole transmisji danych, który określa możliwości i niezawodność tego samochodowego systemu przesyłania danych cyfrowych.



Wstęp

Jak już opisano w pierwszej części artykułu, CAN jest szeregowym, asynchronicznym systemem komunikacyjnym łączącym czujniki i elementy wykonawcze elektronicznych stacji sterujących w samochodach. Wśród wielu jego funkcji główną jest przesyłanie danych cyfrowych. Jest to system asynchroniczny, ponieważ każda stacja (nazywana także „węzłem”) jest synchronizowana przez wiadomość z innej stacji, z boczem wiodącym pierwszego bitu wiadomości (komunikatu), a także następnych wiodących zboczy pozostałej części wiadomości. Zdolność każdej stacji do synchronizowania innej stacji jest określona przez maksymalną różnicę częstotliwości ich oscylatorów. Innymi czynnikami są, na przykład, czas trwania bitu, czas trwania i struktura wiadomości oraz potwierdzenie odbioru (ang. handshaking). Najważniejsze elementy sieci tworzą jej warstwę fizyczną zawierającą topologię sieci i podłączenia do magistrali oraz warstwę przesyłania danych, która określa, ja-

kie medium transmisji danych jest dostępne, jaka jest struktura wiadomości (adres, dane, kontrola i zabezpieczenia przed błędami) i jaki jest protokół transmisji danych.

Zastosowanie

Wymiana informacji między dwoma stacjami sieci może odbywać się dwoma sposobami: przez odwołanie się do określonej stacji (zorientowanie na stację) lub przez podanie określonej wiadomości (zorientowanie na wiadomość).

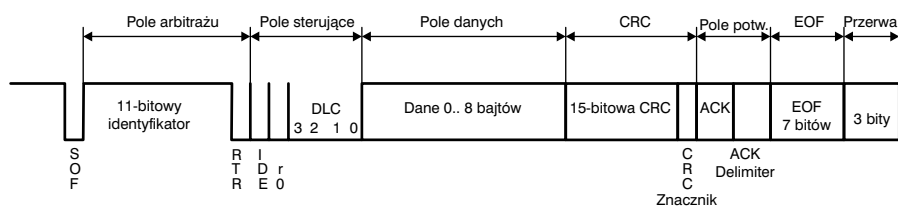
Adresowanie stacji

W tym trybie nadawca adresuje odbiornik podając po prostu adres odbiornika, na przykład: „Stacja 25 przesyła wiadomość do stacji 37”. W ten sposób ustalone jest rzeczywiste połączenie między nadajnikiem (wysyłającym) i odbiornikiem (odbierającym).

Dlatego transmitowany pakiet danych zawiera adres stacji odbiorczej, a także stacji nadającej. Pozostałe stacje dołączone do magistrali ignorują ten pakiet ponie-

Artykuł publikujemy na podstawie umowy z wydawcą miesięcznika „Elektor Electronics”.

Editorial items appearing on pages 21..24 are the copyright property of (C) Segment B.V., the Netherlands, 1998 which reserves all rights.



Rys. 6. Struktura ramki danych (format ramki standardowej – CAN 20A).

waż nie jest on do nich adresowany.

Stacja odbierająca ocenia wiadomość i zazwyczaj potwierdza jej odbiór. W przypadku wystąpienia błędu podczas transmisji danych (brak potwierdzenia z odbiornika), stacja wysyłająca powtarza wiadomość.

Wymiana określona wiadomością

W tym trybie stacja nadająca dodaje do wiadomości niepowtarzalny identyfikator i wysyła wiadomość wraz z tym identyfikatorem przez magistralę, na przykład: „Stacja A przesyła wyniki pomiaru napięcia z identyfikatorem 978“. W tym trybie adresy stacji nadawczej i odbiorczej nie są dołączane do wiadomości.

Taka wiadomość jest oczywiście przeznaczona do kilku stacji odbiorczych dołączonych do magistrali, które korzystają z niej zgodnie z dewizą: „Pobieraj z magistrali to co jest ci potrzebne“. Wówczas kilka stacji musi określić za pomocą swojego oprogramowania, czy wiadomość jest odpowiednia dla nich czy nie.

Przepływ wiadomości

Przepływ wiadomości między indywidualnymi stacjami dołączonymi do magistrali CAN jest realizowany przez nadawanie w warunkach kontrolowanej rywalizacji opatrzonej odpowiednim priorytetem wiadomości lub ramek.

Dominujące i recesywne stany magistrali lub bitów

Rzeczywista transmisja danych przez medium transmisyjne danych nie odbywa się jak zwykle w postaci jedynek i zer, ale przez bity dominujące i recesywne (ustępujące, dominowane). Stan recesywny jest rodzajem stanu magistrali, który może być „przykrywany“ (nadpisywany) przez stan dominujący magistrali. W ten sposób, gdy stacja dołączona do

magistrali wysyła recesywny (ustępujący, dominowany) bit, a inna stacja w tym samym czasie wysyła bit dominujący, to bit dominujący pierwszeństwo przed bitem recesywnym, co oznacza, że stan dominujący jest akceptowany przez całą magistralę. Przyporządkowanie stanów logicznych na magistrali jest generalnie takie, że wartość logiczna zero reprezentuje stan dominujący, a wartość logiczna 1 stan recesywny.

To ustalenie tworzy podstawy specyfikacji CAN i będzie dokładniej opisane dalej.

Pakiety danych

Do wymiany danych przez magistralę, w sieci używane są cztery rodzaje pakietów danych nazywanych zazwyczaj ramkami: ramka danych, ramka zdalnego wywołania, ramka sygnalizacji błędu i ramka przepełnienia.

Ramka danych

Ramka danych jest stosowana przez stacje aby zgodnie z ich oprogramowaniem przesłać dane w linię. Format typowej ramki, która składa się z pojedynczych pól pokazano na **rys. 6**. Jest to format ramki standardowej, zgodny ze specyfikacją systemu CAN2.0A. Znaczenie poszczególnych elementów składowych ramki podanych na rysunku jest następujące:

SOF. Jest to bit startowy ramki, który jest zawsze bitem dominującym (0). Wszystkie stacje dołączone do magistrali synchronizują swoje wewnętrzne stopnie odbiorcze z narastającym zboczem tego bitu (impulsu).

Pole arbitrażu (decyzyjne). To pole, o długości 12 bitów, zawiera dane określające dostęp do magistrali.

Identyfikator 11-bitowy. To pole zawiera identyfikator (ID) transmitowanych ramek. Słowo 11-bitowe umożliwia utworzenie aż do $2^{11} = 2048$ różnych iden-

fikatorów, z których dostępnych jest tylko 2032: pozostałych 16 jest zarezerwowanych dla specjalnych funkcji. To oznacza, że pojedynczy sterownik sieci może przetworzyć 2032 różnych wiadomości (wartości zmierzone, pozycja przełączników, funkcje sygnalizacyjne itp.). Chociaż wydaje się, że jest to duża liczba, to w wielu zastosowaniach nie jest wystarczająca. Dlatego też został opracowany format ramki rozszerzonej EFF (ang. Extended Frame Format) z identyfikatorem 29-bitowym (CAN 20B). W tym standardzie może być przetworzonych $2^{29} = 536\,870\,912$ ramek.

Bit zdalnego żądania transmisji RTR (ang. Remote Transmission Request). Ten bit, który jest zwykle dominującym (0), umożliwia stacji zaadresowanie i wysłanie wiadomości do innej określonej stacji. Jest to bardzo ważne, gdy jakieś dane są pilnie potrzebne do przetworzenia (więcej na ten temat dalej).

Pole kontrolne Jest to 6-bitowe pole zawierające informację jak zbudowana jest ramka danych.

Bit rozszerzenia identyfikatora IDE (ang. Identifier Extension). Wartość tego bitu wskazuje czy jest transmitowana ramka w standardowym formacie z identyfikatorem 11-bitowym (bit IDE ń dominujący = 0), czy ramka w formacie rozszerzonym z identyfikatorem 29-bitowym (bit IDE ń recesywny = 1).

Bit r0 (bit rezerwowany 0). Ten dominujący bit został przewidziany jako zapasowy dla ewentualnego rozszerzenia specyfikacji systemu.

DLC (ang. Data Length Code). To 4-bitowe pole wskazuje ile bajtów danych jest kolejno transmitowanych w polu danych. Specyfikacja systemu CAN określa długość pola danych na 0..8 bajtów, to oznacza, że w pojedynczej ramce danych może być transmitowanych nie więcej niż 8 bajtów danych.

Pole danych. To 8-bajtowe pole zawiera bajty transmitowanych danych (0..8).

Pole CRC. Pole CRC o długości 15 bitów zawiera dodatkowe informacje wprowadzone w celu zabezpieczenia transmitowanych danych przed błędami. W tym celu

stacja nadająca tworzy, zgodnie z określonymi zasadami, 15-bitową sumę kontrolną CRC na podstawie wysyłanych danych i wysyła ją razem z ramką danych. Stacja odbierająca oblicza, zgodnie z tymi samymi zasadami, na podstawie odebranych danych podobną sumę kontrolną i porównuje ją z odebraną. Jeżeli wartości tych dwóch sum są identyczne (zwykły przypadek), to transmisję danych można kontynuować. Jeżeli sumy nie są identyczne, to uruchamiana jest procedura korekcji błędów. Pole CRC jest kończone bitem ogranicznika, który jest zwykle transmitowany w postaci recesywnej.

Pole potwierdzenia. To 2-bitowe pole potwierdzenia służy do wysłania potwierdzenia poprawności odebrania ramek danych.

Przerwa ACK. To 1-bitowe pole jest transmitowane w postaci recesywnej i dlatego może być „przykryte” (nadpisane) bitem dominującym transmitowanym przez inną stację dołączoną do magistrali. Umożliwia to stacjom odbierającym wysłanie potwierdzenia odebrania poprawnej ramki danych. Bit potwierdzenia jest bitem dominującym i jest transmitowany przez stację, zawsze po odebraniu wiadomości wolnych od błędów. Ponieważ jest to bit dominujący, to „przykrywa” bit recesywny wysyłany przez stację nadającą. A zatem, jeżeli stacja nadająca odbiera bit dominujący podczas okienka przerwy ACK, zamiast swojego własnego, wysłanego wcześniej bitu recesywnego, to jest informowana, że przynajmniej jedna stacja odebrała wiadomość.

Okienko przerwy ACK jest zakończone transmitowanym również recesywnie bitem ogranicznika ACK (ang. ACK Delimiter).

Pole zakończenia ramki EOF (ang. End of Frame). To pole składa się z siedmiu recesywnych bitów i kończy ramkę danych.

Przed następną ramką danych, która może być transmitowana, stacje odbierające potrzebują krótkiej przerwy, która umożliwia im przetworzenie lub przynajmniej zapamiętanie odebranych danych. Przerwa ta (ang. Intermission) jest określona przez trzy recesywne bity pola przerwy kończącego ramkę danych.

Z powodu braku miejsca, format ramki rozszerzonej (EFF) nie będzie tu omawiany; zasady jej tworzenia są takie same jak dla formatu ramki standardowej (SFF).

Unikanie konfliktów

Ponieważ wszystkie stacje są dołączone do jednej magistrali CAN, to pojawiają się dwa problemy, które należy rozwiązać:

- Co stanie się, gdy kilka stacji zechce wysłać wiadomość w tym samym czasie?
- Jak jest podejmowana decyzja, która stacja może rozpocząć nadawanie, a która stacja musi poczekać ze swoją transmisją?

Oczywiście, nie rozwiązanie tych kwestii może prowadzić do konfliktów i w celu ich uniknięcia stosowana jest specjalna procedura dostępu do magistrali, która musi być przestrzegana przez wszystkie stacje, gdy chcą wysłać wiadomości. W tej procedurze ważną rolę odgrywają właśnie bity dominujące i recesywne w polu arbitrażowym (ang. Arbitration Field).

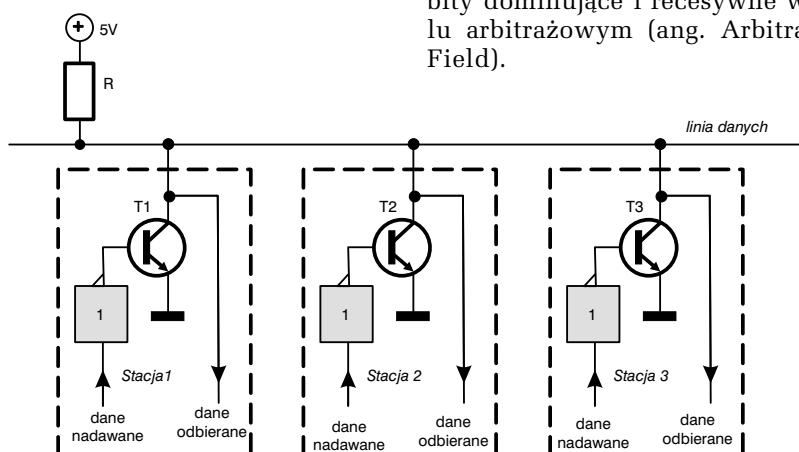
Tab. 3. Porównanie parametrów systemów CAN 20A (format ramki standardowej) i CAN 20B (format ramki rozszerzonej)

| 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 |

Zasadniczo każda stacja nadająca „słyszy” swoje własne przesłanie na magistralę: wysyła jakiś bit, odbiera go z powrotem i porównuje z wysłanym. Jeżeli te dwa bity są identyczne, to transmisja wiadomości jest dozwolona. Jeżeli jednak te dwa bity nie są jednakowe, to jest problem. Wspomniany wcześniej bit recesywny (o wartości 1) może być „przykryty” przez bit dominujący (0).

Na rys. 7 pokazano, w dużym uproszczeniu, konfigurację obwodów dołączających stacje do magistrali (stopnie dołączające stacje do magistrali). W zasadzie są to stopnie wyjściowe w konfiguracji z otwartym kolektorem, które tworzą połączenie iloczynu galwanicznego (Wired - AND, zwarte AND). W odniesieniu do stacji 1 transmitowany recesywnie bit 1 zapewnia (gwarantuje), że tranzystor T1 pozostaje odcięty (nie przewodzi). To oznacza, że poziom recesywny jest wstępnie ustawiony na magistrali. Po wysłaniu tego bitu stacja 1 odczytuje stan magistrali i określa bit jaki został wysłany. Jeżeli później transmitowany jest dominujący bit (0), to tranzystor T1 zostaje włączony i zzwiera linię magistrali do masy. Linia magistrali jest zatem w stanie dominującym (0). Ponownie stacja 1 odczytuje zwrotnie bit, który wysłała. Dla tych trzech stacji, jeżeli jedna z nich wysyła bit dominujący, to stan linii magistrali staje się dominujący (0) i inne stacje odczytują ten poziom.

Jak jest realizowana procedura dostępu do magistrali pokażemy na przykładzie. Przyjmijmy, że wszystkie stacje z rys. 7 są gotowe do transmitowania swoich ramek danych z trzema różnymi identyfikatorami:



Rys. 7. Bardzo uproszczony schemat obwodów wejściowo/wyjściowych stacji, ilustrujący sposób ich dołączenia do magistrali CAN.

Stacja 1: identyfikator 367;
 Stacja 2: identyfikator 232;
 Stacja 3: identyfikator 239.

Wszystkie trzy rozpoczynają uzgadnianie dostępu do magistrali fazą arbitrażową wysyłając bit SOF (patrz rys. 8). Jest to bit dominujący i każda stacja odczytuje zwrotnie swój własny (okazuje się, że poprawny) bit z magistrali. Następnie wysyłane są identyfikatory. W czasie b wszystkie stacje wysyłają bit dominujący i wszystko jest w porządku. W czasie c w dalszym ciągu nie ma problemów. W czasie d stacja 1 wysyła recesywny bit (1), podczas gdy stacje 2 i 3 kontynuują transmisję z bitami dominującymi (0). Po odczycie zwrotnym wysłanego bitu, stacja 1 „zauważa”, że wysłany przez nią bit został nadpisany bitem dominującym, co oznacza, że utraciła ona dostęp do magistrali na rzecz przynajmniej jednej z pozostałych stacji. W tej sytuacji stacja 1 wchodzi w tryb odbioru (choć próbuje znowu wysłać wiadomość później). Natomiast stacje 2 i 3 kontynuują rozpoczętą transmisję jak poprzednio.

W czasie j stacja 3 wysyła recesywny bit, który jest natychmiast nadpisywany bitem dominującym wysyłanym przez stację 2. Jest to „zauważane” przez stację 3, która wskutek tego przestawia się na odbiór (podobnie jak stacja 1, spróbuje wysłać wiadomość później). W tym uzgadnianiu dostępu stacja 2 zwyciężyła i może wysyłać swoje wiadomości na magistralę bez dalszych przeszkód.

Porównując identyfikatory stacji widzimy, że pierwsza uzyskała dostęp do magistrali stacja, która ma najmniejszy numer identyfikatora: ma ona najwyższy priorytet wysyłania wiadomości. Inaczej,

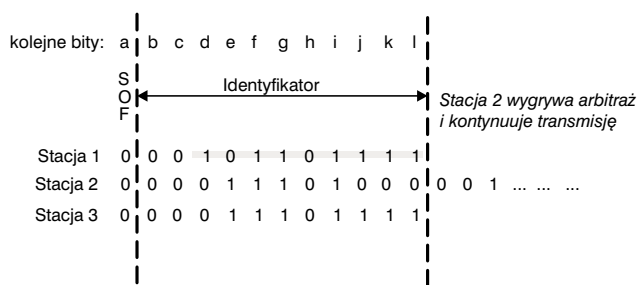
w liczbie identyfikatora zawarte jest także automatycznie pierwszeństwo w przesyłaniu wiadomości. Wiadomość z identyfikatorem 0 będzie zawsze wysyłana jako pierwsza przez stacje dołączone do magistrali, ponieważ ma ona najwyższy priorytet. Wiadomość z identyfikatorem 2032 musi natomiast długo oczekiwać, ponieważ ma najniższy priorytet.

Ramka zdalnego żądania transmisji

Ramka zdalnego żądania transmisji spełnia jedną z ważniejszych funkcji w sieci. Przyjmijmy, że stacja D dołączona do magistrali CAN wysyła co pięć minut dane trzech wartości zmierzonej temperatury z identyfikatorem 598. To oznacza, że pole danych zawiera trzy bajty. Te wiadomości są odbierane i przetwarzane przez inne stacje.

Jednakże stacja G pilnie potrzebuje wartości aktualnie zmierzonej temperatury i nie może w żadnym wypadku czekać na transmisję przez pięć minut. Dlatego może zażądać wyników pomiarów bezpośrednio ze stacji D, to jest może „obejść” normalny cykl transmitowania danych. Aby to zrealizować stacja wysyła ramkę zdalnego żądania transmisji (RRF), która ma strukturę podobną jak ramka danych DF (rys. 6), jednak z kilkoma małymi różnicami:

- Identyfikator stacji, do której jest wysyłane żądanie (tutaj 598) jest podawany w polu identyfikatora.



Rys. 8. Diagram ilustrujący ustalanie przez stacje dostępu do magistrali (arbitraż).

- Liczba użytecznych bajtów zawartych w wywoływanej wiadomości (tutaj 3) jest podawana w polu DLC.
- Bit zdalnego żądania transmisji RTR (Remote Transmission Request), który jest bitem dominującym (0) w ramce danych jest tworzony i transmitowany recesywnie (1). Jest to typowy sposób identyfikacji stacji, która żąda danych bezpośrednio z innej określonej identyfikatorem stacji.
- Nie ma pola danych w ramce zdalnego żądania transmisji (RRF): pole DLC jest bezpośrednio przed polem CRC. Inaczej mówiąc, ramka zdalnego żądania transmisji (RRF) jest skomponowana podobnie do ramki danych, ale z liczbą 0 bajtów danych.

Realizacja funkcji zdalnego żądania transmisji przebiega następująco. Wszystkie stacje dołączone do magistrali odbierają ramkę i rozpoznają wskutek ustawienia bitów w RTR, że jakaś stacja zażądała określonych danych od innej stacji. Stacja D ustala, że identyfikator w ramce zdalnego żądania transmisji jest taki sam jak jej własny identyfikator i natychmiast przesyła swoją odpowiedź w postaci ramki z żadanymi danymi.

EE

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.

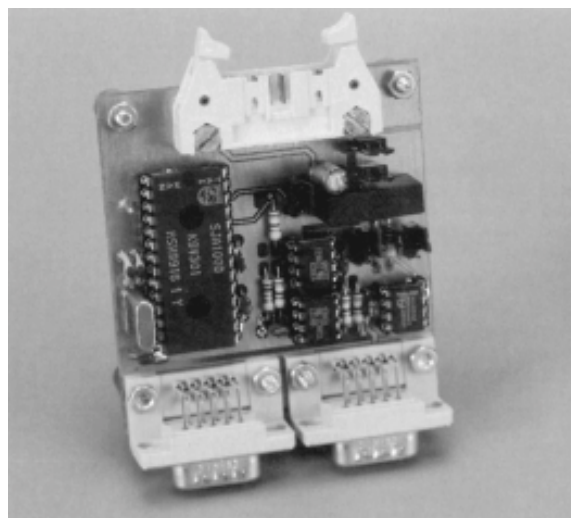
Magistralą CAN dane mogą być transmitowane z szybkością 500 kbitów/s. Na każde 0,7 s przypada jeden błędny bit spowodowany zewnętrznymi zakłóceniami. Sieć może pracować osiem godzin dziennie przez 365 dni w roku. Wbudowany sposób zabezpieczenia przed błędami gwarantuje, że przez 1000 lat pracy tylko 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.

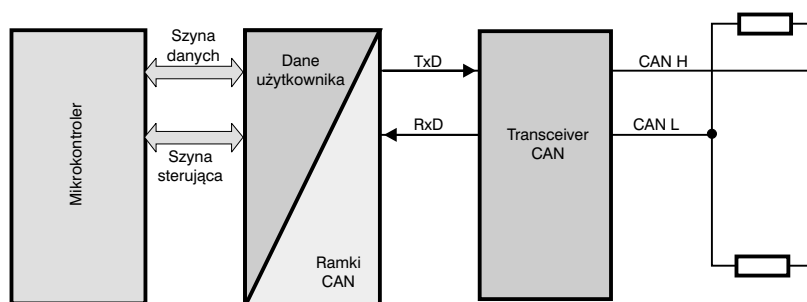
Jeżeli stacja nadająca stwierdzi, że jej bit w przerwie na potwierdzenie (ACK slot) nie został nadpisany, to „wie”, że żadna stacja nie odebrała jej wiadomości poprawnie. W takim razie, zawieszając swoje dotychczasowe działanie i wywołuje procedurę korekcji błędów (zobacz dalej).

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 w 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

Magistrala CAN, część 4

Interfejs magistrali CAN

W tej części artykułu przedstawiamy konstrukcję interfejsu magistrali CAN, wykonanego w oparciu o układ scalony SJA1000 firmy Philips.

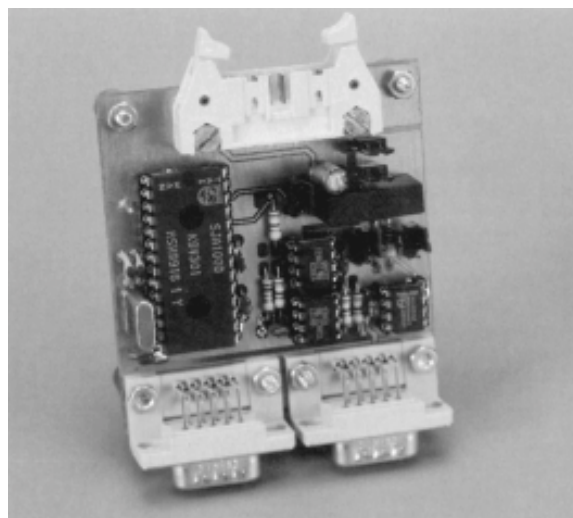
Artykuł publikujemy na podstawie umowy z wydawcą miesięcznika "Elektor Electronics".

Editorial items appearing on pages 13..15 are the copyright property of (C) Segment B.V., the Netherlands, 1998 which reserves all rights.

Po długich rozważaniach na temat właściwości interfejsu magistrali CAN można teraz przystąpić do opisanego jego budowy. Schemat interfejsu przedstawiono na rys. 10, a widok płytki drukowanej na rys. 11. Podstawowe parametry scalonego sterownika magistrali zostały zamieszczone w tab. 4 (EP3/2000).

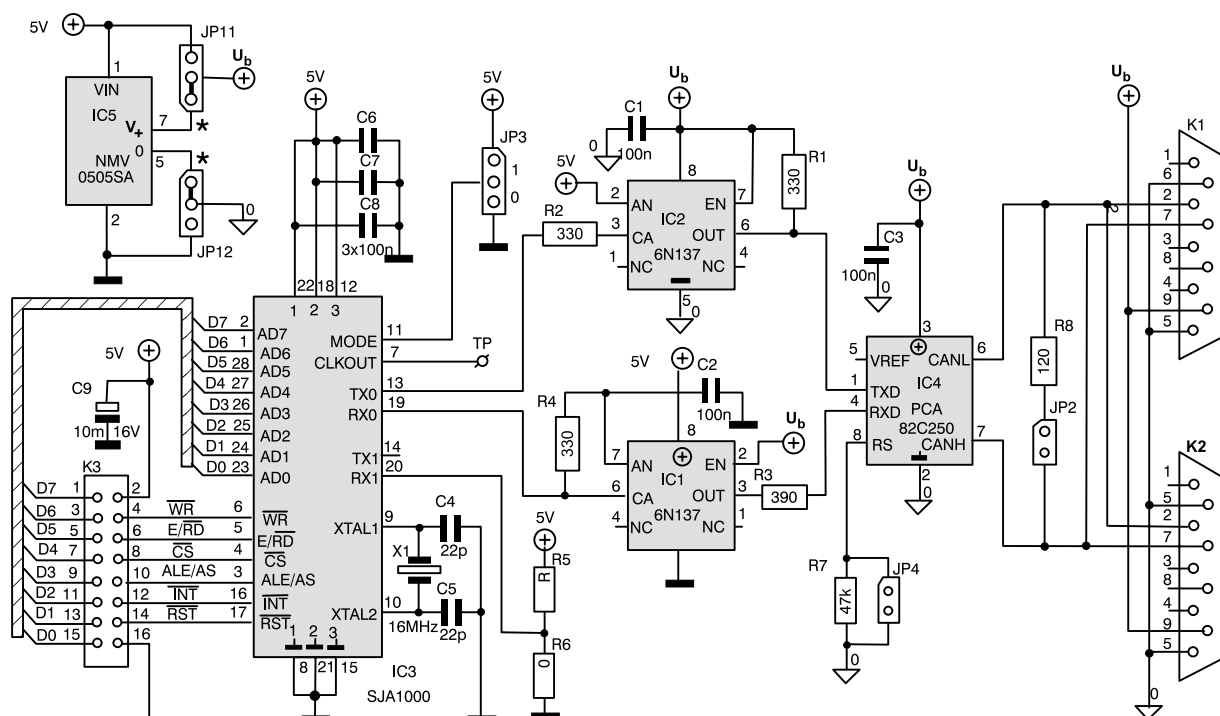
Sterownikiem CAN jest układ scalony SJA1000 (IC3). Jego schemat blokowy pokazano na rys. 12. Układ ten jest następcą PCA82C200, z którym w trybie 1 jest zgodny w zakresie sprzętowym i programowym, a także rozmieszczenia wyprowadzeń.

Interfejs może być używany do współdziałania z mikrosterownikiem Motoroli albo Intela lub innym z nimi zgodnym. Rolę



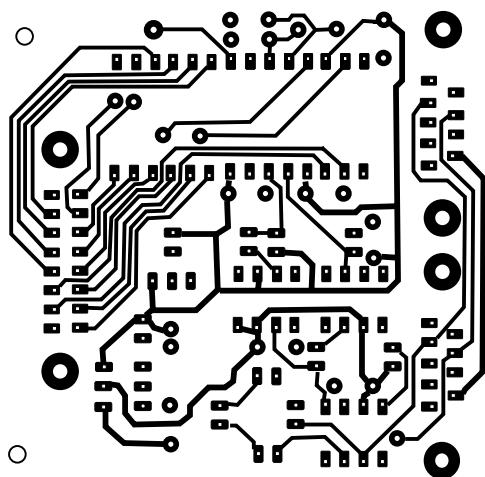
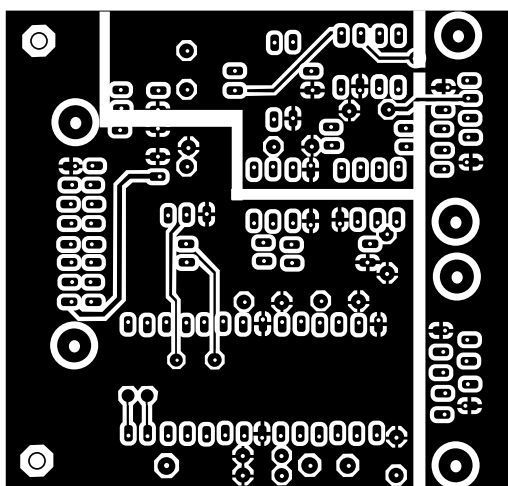
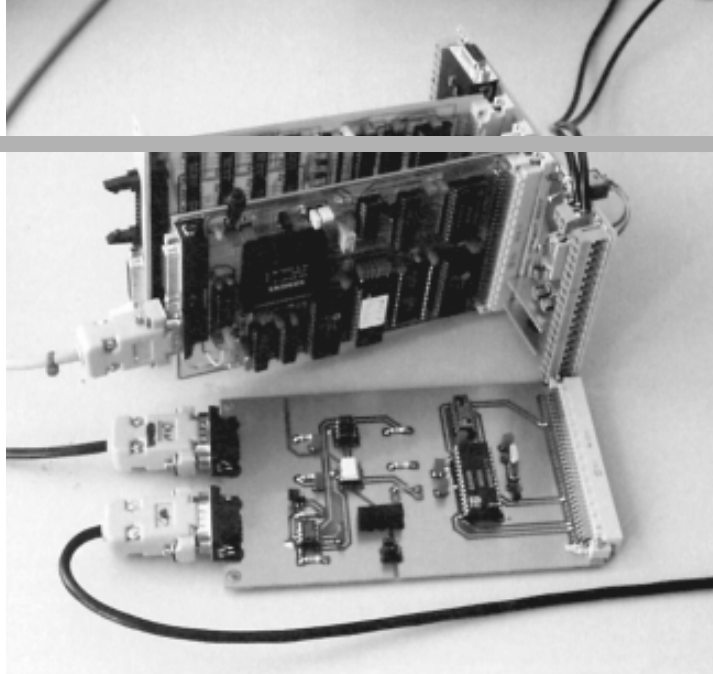
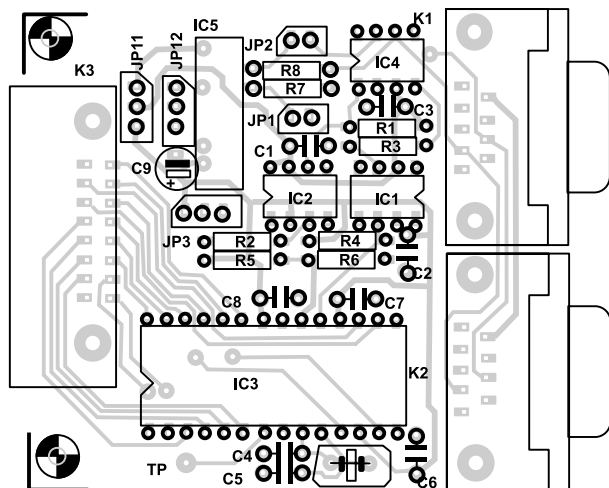
transceivera CAN spełnia układ IC4, oznaczony symbolem PCA82C250.

Mikrosterownik łączy się z interfejsem magistrali CAN odcinkiem przewodu taśmowego, który nie powinien być dłuższy od 10cm i jest zakończony 16-stykowym złączem K3. Wyprowadzenia K3 opisano w tab. 5.



Rys. 10. Schemat ideowy interfejsu magistrali CAN.

* zob. tekst



Rys. 11. Płytką drukowana interfejsu magistrali CAN.

Przez przewód taśmowy mikrosterownik wymienia dane operacyjne, dane sterowania i dane stanu ze sterownikiem CAN. Dane te są przetwarzane przez sterownik zarówno w kierunku wysyłania, jak i odbioru. Mikrosterownik „widzi” sterownik CAN w postaci rozszerzenia swojej pa-

mięci, do której wpisuje wysyłane dane operacyjne albo z której odczytuje otrzymywane dane operacyjne.

Częstotliwość sygnału zegarowego, która ulega podziałowi w kilku stopniach, może zostać zmierzona na wyprowadzeniu testującym TP, gdy trzeba się upewnić, czy do sterownika można bezpiecznie uzyskać dostęp i go programować.

Szeregowy sygnał wyjściowy jest pobierany z wyprowadzenia 13 sterownika i przez optoizolator IC2 doprowadzany do końcówki 1 transceivera. Transceiver generuje standardowe sygnały magistrali CAN, które z jego końcówek 6 i 7, przez złącza K1 i K2, są wysyłane nie ekranowaną skrętka miedziana.

Sygnał odebrany z magistrali pojawia się na końcówce 4 transeivera, z której poprzez optoizolator IC1 jest przesyłany do końcówki 19 sterownika. Sterownik przetwarza odebrane bity zgodnie ze stosownym protokołem CAN. Sygnał ten jest następnie przekazywany do mikrosterownika do analizy.

Optoizolatory IC1 i IC2 oraz przetwornik 5V DC/DC (IC5) izolują sekcje mikrosterownika i magistrali od węzła (stacji). Rozwiązanie to zapewnia, że żaden błąd nie ekranowanej skretki, pomimo doj-

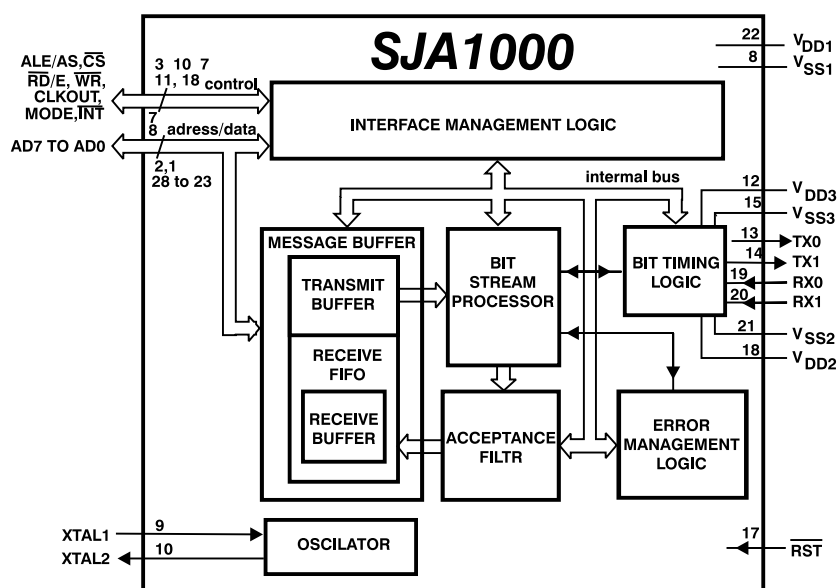
ścia do transceivera, nie może uszkodzić sekcji mikrosterownika i związanego z nim systemu.

Można oczywiście wykonać interfejs bez stopni izolujących, czyli pominąć R1..R4, C1..C2, IC1, IC2 i IC5 oraz JP2. Końcówki zasilania, a także końcówki wejściowe i wyjściowe IC3 i IC4 muszą być wtedy odpowiednio ze sobą połączone. Trzeba jednak pamiętać, że sygnały zakłóceń ze skrętki będą wtedy bez przeszkód mogły dochodzić do sekcji mikrosterownika.

Zworki JP11 i JP12 (oznaczone na płycie przez „*“) ustalają sposób zasilania interfejsu i mikrosterownika. Przy ustawieniu tak jak na rys. 10, zapewniona jest izolacja IC1, IC2 i IC4 od zewnętrznego zasilania i układy te są zasilane przez IC5. Przy przeciwnym położeniu zworek izolacji nie ma i wszystkie stopnie są

Tab. 5. Styki K3.

| Styk | Połączenie | Funkcja |
|------|------------|-----------------------------|
| 1 | D7 | Dane I/O |
| 3 | D6 | Dane I/O |
| 5 | D5 | Dane I/O |
| 7 | D4 | Dane I/O |
| 9 | D3 | Dane I/O |
| 11 | D2 | Dane I/O |
| 13 | D1 | Dane I/O |
| 15 | D0 | Dane I/O |
| 2 | + 5V | Zasilanie (+) |
| 4 | WR\ | Sygnał Write\ |
| 6 | RD\ | Sygnał Read\ |
| 8 | CS\ | Sygnał Chip Select |
| 10 | ALE | Sygnał Address latch enable |
| 12 | INT\ | Sygnał Interrupt\ |
| 14 | RST\ | Sygnał Reset\ |
| 16 | GND | Masa |



Rys. 12. Wewnętrzny schemat blokowy układu scalonego sterownika CAN, SJA1000.

zasilane bezpośrednio z końcówek 0 i +Ub.

Alternatywnym rozwiązaniem jest dostarczenie napięcia zasilającego przewodami poprowadzonymi równolegle do skrętki i pobieranie go z końcówek 6 i 9 odpowiednio K2 i K1.

Wskutek użycia zwory Jp2, następuje przyłączenie do końcówek 6 i 7 transceivera rezystora obciążającego (terminującego) magistrali R8. Trzeba jednak pamiętać, że do magistrali mogą być przyłączone tylko dwa rezystory terminujące, jeden na początku, a drugi na końcu skrętki. Dalsze rezystory zakończone (w innych węzłach) łączyłyby się równolegle z tymi dwoma, zmniejszałyby całkowitą oporność obciążenia, zwiększając prąd obciążenia transceivera, co mogłoby doprowadzić do jego termicznego przeciążenia, a nawet zniszczenia.

Zwornę JP3 należy ustawić zależnie od użytego mikrosterownika. Pozycja pokazana na rys. 10 (napięcie + 5V doprowadzone do końcówki 11, oznaczona na płytce znakiem „1“, służy procesorom Intela lub z nim zgodnym).

Usunięcie tej zworki (i zwarcie końcówki 11 IC3 z masą) jest stosowane do procesora Motoroli lub z nim zgodnych.

Od położenia zworki JP4 zależy nachylenie zboczy impulsów w magistrali CAN. W przypadku dużych szybkości transmisji danych (do 1Mb/s) zachowanie stromego nachylenia zboczy jest bardzo ważne, ale zwiększa to emisję zakłóceń, wywoływanych przez impulsy CAN. Zakłócenia te można stłumić tylko przez użycie skrętki ekranowanej. W tym przypadku trzeba użyć zworki zwierającej rezystor R7.

WYKAZ ELEMENTÓW

Rezystory

R1..R4: 390Ω
R5, R6: zob. tekst
R7: 47kΩ
R8: 120Ω

Kondensatory

C1..C3, C6..C8: 0,1μF, ceramiczne
C4, C5: 22pF, ceramiczne
C9: 10μF/16V, stojący

Półprzewodniki

IC1, IC2: 6N137
IC3: SJA1000
IC4: PCA82C250
IC5: NMV505S.A. (Newport/Farnel)

Różne

X1: oscylator kwarcowy 16MHz
K1, K2: złącze D-9, wyprowadzenia pod kątem prostym, do druku
K3: header 16-stykowy z zatraskiem, wyprowadzenia pod kątem prostym, do druku
JP2, JP4: 2 styki z listwy szpilkowej 2,54 mm ze zworną
JP3, JP11, JP12: 3 styki z listwy szpilkowej 2,54 mm ze zworną

Przy mniejszych szybkościach transmisji (do 125kb/s) zbocza impulsów nie muszą być tak strome, dzięki czemu zakłócenia wywoływane przez impulsy CAN nie są tak silne i można użyć skrętki nie ekranowanej, a R7 nie jest zwierany.

Kończymy opis interfejsu magistrali CAN. Następny artykuł będzie dotyczył połączenia magistrali CAN z mikrosterownikiem i zastosowania magistrali CAN.

EE

Magistrala CAN, część 5

Oprogramowanie interfejsu

Sieć CAN składa się oczywiście nie tylko z opisanego w poprzedniej części artykułu interfejsu magistrali. Interfejs jest po prostu łącznikiem pomiędzy mikrosterownikiem czy komputerem a właściwą magistralą CAN. Układ, szczególnie opisany przed miesiącem, do prawidłowego działania wymaga oprogramowania sterującego i to właśnie jest tematem tej części publikacji.



Każda stacja lub węzeł systemu magistrali CAN wymaga, poza interfejsem magistrali, mikrosterownika lub komputera z odpowiednim programem. Do uruchomienia stacji, jej sprawdzenia i obsługiwania potrzeba dwóch zestawów programów: oprogramowania operacyjnego i oprogramowania aplikacyjnego.

Program operacyjny wprowadza cały system w ruch i zapewnia jego działanie, a także testuje interfejs wraz z mikrosterownikiem lub komputerem. Taki test wykazuje, czy sterowanie przez mikrosterownik/komputer działa właściwie, tak z punktu widzenia sprzętowego, jak i programowego, oraz czy dane są poprawnie przesyłane do magistrali CAN. Test więc pomaga w ustanowieniu prostej ścieżki łączności pomiędzy dwoma lub kilkoma węzłami.

Oprogramowanie aplikacyjne jest związane ze szczególną rolą mikrosterownika/komputera w sieci. Od niego zależy konkretne zastosowanie stacji: rejestracja pomiarów, sterowanie wyświetla-

czem, transmisja czasu i daty czy jeszcze inne cele.

Do działania każdego węzła jest więc potrzebny szczególny program, odpowiedni do jego funkcji. Na działanie sieci składa się suma wszystkich funkcji wykonywanych przez poszczególne stacje. Innymi słowy, w celu osiągnięcia oczekiwanych rezultatów przestrzennie rozmieszczona sieć może być sterowana i monitorowana jako całość.

Program operacyjny

Programowanie sterownika CAN podlega tym samym ogólnym zasadom, jak zewnętrznych urządzeń peryferyjnych:

- Funkcja sterownika jest stała albo ustawiana (zaprogramowana) przez zestawy danych zapisywane w Rejestrach Funkcji Specjalnych (SFR).
- Wewnętrzne SFR-y są przez mikrosterownik/komputer interpretowane jako normalne adresy pamięci w zakresie zewnętrznej RAM, pod którymi dane mogą być wpisywane lub odczytywane. Oznacza to, że mikrosterow-

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

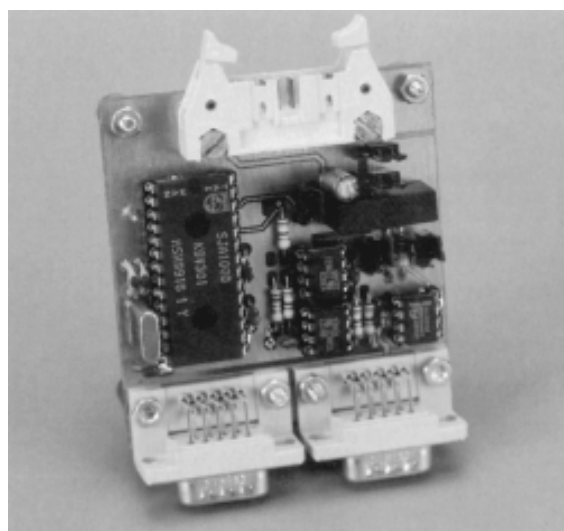
Editorial items appearing on pages 13..16 are the copyright property of (C) Segment B.V., the Netherlands, 1998 which reserves all rights.

nik/komputer nie wie, że działa wraz ze sterownikiem CAN. Istotny z jego punktu widzenia i dla oprogramowania aplikacyjnego jest jedynie dostęp do określonych miejsc w pamięci. Zatem przy tworzeniu oprogramowania aplikacyjnego dla sterownika IC3 należy wykonać następujące zadania:

- ustalić podstawowy adres selekcyjny dla sterownika SJA1000,
- zinterpretować wewnętrzne ustawienia struktury SFR-ów w sterowniku,

- stworzyć procedurę podstawowej inicjalizacji sterownika,
- stworzyć procedurę dostarczania danych do magistrali,
- stworzyć procedurę odbioru danych z magistrali CAN.

Opis realizacji tych zadań dla podstawowego trybu CAN sterownika zostanie przedstawiony pokrótce w następnych paragrafach. Obszerniejsze i bardziej szczegółowe informacje można znaleźć w danych technicznych i notach aplikacyjnych sterownika.



Znaczenie poszczególnych kolumn tabeli jest następujące:

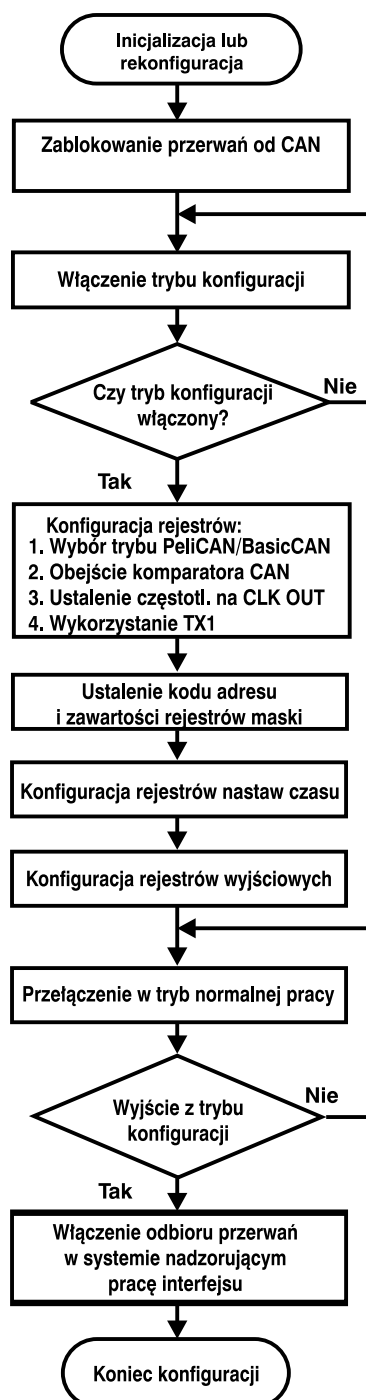
1. W pierwszej kolumnie, adresów CAN, znajdują się wewnętrzne adresy odpowiednich SFR-ów, do których trzeba tylko dodać podstawowy adres wyboru układu. Jeżeli, na przykład, potrzebny jest dostęp do rejestru stanu sterownika, to do wewnętrznego adresu SFR-a, wynoszącego 2, trzeba dodać F000H. Jeżeli zatem wykonywana jest na tym rejestrze operacja odczytu lub zapisu, program musi udostępnić adres F002H w zewnętrznej RAM. Od tego momentu rejestr dzielnika sygnału zegarowego będzie dostępny pod adresem F001FH (=F000H+31D=F000H+1FH - uwaga, użyto dwóch różnych systemów liczbowych).

2. Druga kolumna pokazuje podział SFR-ów na trzy różne grupy: grupę sterowania, grupę bufora nadawania i grupę bufora odbioru.

3. Sterownik funkcjonuje w dwóch, sterowanych programem, trybach:

- w trybie działania, będącym normalnym trybem pracy,
- w trybie kasowania, będącym trybem IC3 w czasie kasowania sprzętowego albo gdy bit kasowania rejestru sterowania jest ustawiony. Sterownik powraca wtedy do normalnego trybu działania.

Tryb kasowania jest potrzebny do inicjalizacji sterownika, tylko w tym trybie bowiem mogą zostać ustawione niektóre parametry



Rys. 1.

Ustalanie podstawowego adresu selekcyjnego

Dostęp do układu jest możliwy poprzez podstawowy adres selekcyjny (wyboru układu). Ponieważ sterownik IC3, w podstawowym trybie CAN, wymaga spójnego 32-bajтового zakresu adresów zewnętrznych, a w trybie PeliCAN jednego ze 128 bajtów, maksymalny zakres został ustalony na 128 bajtów, aby nie wykluczać możliwości użycia w przyszłości trybu PeliCAN.

Stan niski na końcówce CS (3) jest dla sterownika SJA1000 sygnałem zezwalającym. Oznacza to, że mikrosterownik/komputer musi kodować adres układu w taki sposób, aby w spójnym zakresie adresowym, nie mniejszym od 128 bajtów, wywołać sygnał stanu niskiego na końcówce 8 złącza K3, co umożliwi sterownikowi CAN przesłanie danych. Pierwszy taki adres staje się tak zwanym podstawowym adresem wyboru chipu sterownika. Gdy mikrosterownik lub komputer sięgnie do mieszczącej się w tym zakresie adresowej pozycji w RAM, uzyskuje bajt zawarty w SFR sterownika lub może do SFR wpisać nowy bajt. Przyjęto, że podstawowym adresem wyboru układu sterownika SJ1000 jest F000H.

Struktura wewnętrzna SFR

Najważniejsze SFR-y sterownika IC3 do pracy w podstawowym trybie CAN zestawiono w tab. 6.

działania. Zostaje wówczas ustawiony bit kasowania (sterownik ustawia swój normalny tryb działania), po czym odpowiednie parametry mogą zostać zmienione i bit kasowania jest wyłączony. Sterownik podejmuje wtedy na nowo działanie ze zmienionymi parametrami.

4. W trzeciej i czwartej kolumnie pokazano:

- funkcje rejestru,
- znaczenie wpisywanej w trybie działania zawartości rejestru,
- znaczenie odczytywanej zawartości rejestru.

5. W piątej i szóstej kolumnie są pokazane odnośne dane rejestru w trybie kasowania.

Przykład opisu SFR o adresie 4

Tryb działania (normalne funkcjonowanie sterownika):

- odczyt - chociaż odczyt tego rejestru jest możliwy, wyniki odczytu nie są użyteczne, ponieważ zawsze odczytuje się wartość FFH.
- zapis - do rejestru nie można niczego wpisać.

Tryb kasowania (sterownik jest w trybie kasowania):

- odczyt - wynikiem odczytu z rejestru jest kod akceptacji,
- zapis - do rejestru można wpisać nowy kod akceptacji.

Z tego przykładu widać, że w czasie normalnego działania sterownika ten SFR nie ma specjalnej funkcji. Trzeba jednak zwrócić uwagę na to, że w trybie kasowania kod akceptacji, z którym sterownik funkcjonuje w czasie normalnego działania, jest ustawiony.

Tworzenie procedury podstawowej inicjalizacji

Przed rozpoczęciem pracy nad tą procedurą niezbędne jest zapoznanie się z notą aplikacyjną sterownika SJA1000 (AN97076 - dostępna w Internecie). Na 23. stronie tego dokumentu jest przytoczony schemat działań ze szczegółowymi komentarzami o sposobie inicjalizacji sterownika. Trzeba też zapoznać się dokładnie z opisem pojedynczego rejestru, co pozwoli bez trudu dobierać parametry zgodnie z indywidualnymi potrzebami.

Tworzenie procedury przesyłania danych

Jak już wspomniano, większość zadań, potrzebnych przy przesyłaniu danych, przejmuje sterownik SJA1000. Wysłanie bajtu danych do magistrali CAN wymaga jedynie czterech czynności:

- dostarczenia sterownikowi identyfikatora (ID) ramki, która ma zostać wysłana,
- wskazania, ile należy wysłać bajtów (0..8) danych,
- określenia, czy ramka jest ramką zdalnego żądania transmisji (RTR),
- wpisania wymaganych bajtów

danych do bufora nadawania w sterowniku.

I to wszystko! Reszta procesu jest automatycznie wykonywana przez sterownik CAN:

- zestawianie ramki,
- obliczanie sumy CRC (cyklicznej kontroli nadmiarowej),
- dołączenie do ramki pozostałych pól,
- uzyskanie dostępu do magistrali,
- wysłanie ramki,
- sprawdzenie błędów.

Za pośrednictwem rejestru stanu użytkownik otrzymuje komunikaty o powodzeniu lub niepowodzeniu transmisji.

Tab. 6. Wewnętrzne SFR-y sterownika, używane w podstawowym trybie działania interfejsu CAN.

| ADRES CAN | SEGMENT | TRYB DZIAŁANIA | | TRYB KASOWANIA | |
|-----------|--------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| | | WPIS | ODCZYT | WPIS | ODCZYT |
| 0 | stero- wanie | sterowanie | sterowanie | sterowanie | sterowanie |
| 1 | | (FFH) | rozkaz | (FFH) | (FFH) |
| 2 | | stan | - | stan | - |
| 3 | | przerwanie | - | przerwanie | - |
| 4 | | (FFH) | - | kod akceptacji | kod akceptacji |
| 5 | | (FFH) | - | maska akceptacji | maska akceptacji |
| 6 | | (FFH) | - | takt magistrali 0 | takt magistrali 0 |
| 7 | | (FFH) | - | takt magistrali 1 | takt magistrali 1 |
| 8 | | (FFH) | - | sterowanie wyj. | sterowanie wyj. |
| 9 | | test | test; uwaga 2 | test | test; uwaga 2 |
| 10 | bufor nadawania | identyfikator (10 do 3) | identyfikator (10 do 3) | (FFH) | - |
| 11 | | identyfikator (2 do 0) RTR i DLC | identyfikator (2 do 0) RTR i DLC | (FFH) | - |
| 12 | | bajt danych 1 | bajt danych 1 | (FFH) | - |
| 13 | | bajt danych 2 | bajt danych 2 | (FFH) | - |
| 14 | | bajt danych 3 | bajt danych 3 | (FFH) | - |
| 15 | | bajt danych 4 | bajt danych 4 | (FFH) | - |
| 16 | | bajt danych 5 | bajt danych 5 | (FFH) | - |
| 17 | | bajt danych 6 | bajt danych 6 | (FFH) | - |
| 18 | | bajt danych 7 | bajt danych 7 | (FFH) | - |
| 19 | | bajt danych 8 | bajt danych 8 | (FFH) | - |
| 20 | bufor odbioru | identyfikator (10 do 3) | identyfikator (10 do 3) | identyfikator (10 do 3) | identyfikator (10 do 3) |
| 21 | | identyfikator (2 do 0) RTR i DLC | identyfikator (2 do 0) RTR i DLC | identyfikator (2 do 0) RTR i DLC | identyfikator (2 do 0) RTR i DLC |
| 22 | | bajt danych 1 | bajt danych 1 | bajt danych 1 | bajt danych 1 |
| 23 | | bajt danych 2 | bajt danych 2 | bajt danych 2 | bajt danych 2 |
| 24 | | bajt danych 3 | bajt danych 3 | bajt danych 3 | bajt danych 3 |
| 25 | | bajt danych 4 | bajt danych 4 | bajt danych 4 | bajt danych 4 |
| 26 | | bajt danych 5 | bajt danych 5 | bajt danych 5 | bajt danych 5 |
| 27 | | bajt danych 6 | bajt danych 6 | bajt danych 6 | bajt danych 6 |
| 28 | | bajt danych 7 | bajt danych 7 | bajt danych 7 | bajt danych 7 |
| 29 | | bajt danych 8 | bajt danych 8 | bajt danych 8 | bajt danych 8 |
| 30 | | (FFH) | - | (FFH) | - |
| 31 | | dzielnik sygnału zegarowego | dzielnik sygnału zegarowego | dzielnik sygnału zegarowego | dzielnik sygnału zegarowego |

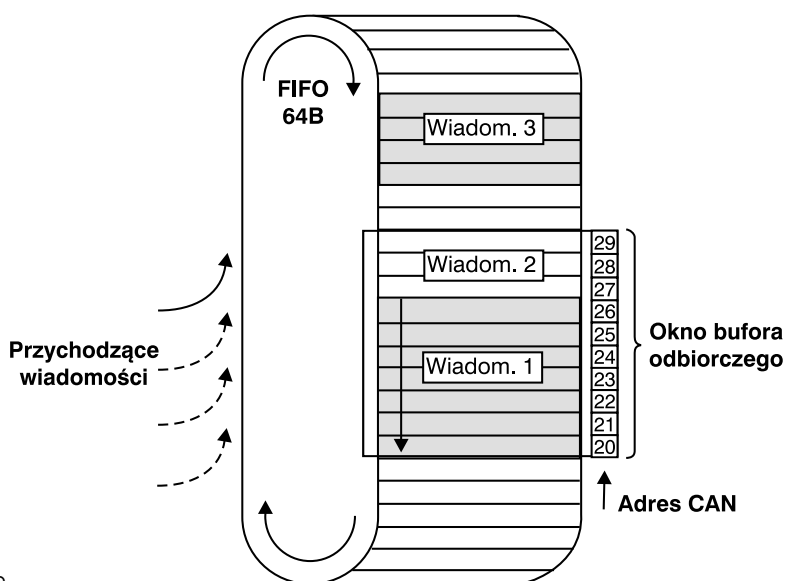
Tworzenie procedury odbioru danych

Przy odbiorze danych sterownik SJA1000 także przejmuje większość zadań, czyli odbiera dane niemal całkowicie automatycznie. Sterownik przetwarza odebrane ramki i wpisuje zawarte w nich potrzebne informacje do sekcji wykrywania błędów i filtru akceptacji w jego RXFIFO. Jeżeli filtr akceptacji jest wyłączony, oceniana jest każda odebrana ramka. W RXFIFO są zapisywane kolejne dane z każdej ramki (tab. 6, adresy 20..29):

- identyfikator ramki,
- bit zdalnego żądania transmisji (RTR),
- kod długości danych (DLC),
- bajty danych użytecznych.

Pojemność wewnętrznej pamięci odbiorczej FIFO w IC3 wynosi dokładnie 64 bajty. Liczba ramek, które mogą być przechowywane w pamięci interfejsu, zależy od rozmiaru ramki, a przede wszystkim od kodu długości danych. Okienkiem bufora odbioru (tab. 6, adresy 20..29), które może być czytane przez użytkownika, jest to, co zostaje przesunięte do okienka przez RXFIFO. Składa się z właśnie odebranego zespołu danych (ramka komunikatu), które użytkownik może przetwarzać za pomocą programu. Łączność pomiędzy SJA1000 a mikrosterownikiem/komputerem w trybie odbioru może przybierać dwie formy:

- *Sterowanie przerwaniem.* Gdy sterownik otrzyma pozbawioną błędów, kompletną ramkę, inicjuje przerwanie w mikrosterowniku poprzez jego końcówkę 16 (INT). Wywołuje to natychmiastową reakcję mikrosterownika/komputera na otrzymany komunikat, który może niezwłocznie zostać odczytany za pośrednictwem sterownika.
- *Operacja odpytywania (polling).* Przy tym rodzaju operacji bit stanu bufora odbioru w rejestrze stanu sterownika jest nieustannie odpytywany przez mikrosterownik/komputer. Gdy bit ten jest ustawiony, co sygnalizuje poprawne odebranie przez sterownik co najmniej jednego komunikatu, program odczytuje ramkę i stosownie ją przetwarza.



Rys. 2.

Po odczytaniu komunikatu program aplikacyjny ponownie udziela zezwolenia okienku bufora odbioru, potwierdzając, że otrzymany komunikat został odebrany i przetworzony. Okienko odbiorcze jest więc gotowe do odbioru z RXFIFO następnej ramki. W ten sposób program aplikacyjny jest informowany o przetworzeniu kolejnych ramek. Trzeba jeszcze wspomnieć o dwóch sprawach:

- Niezwłocznie po odczytaniu i przetworzeniu ramki (komunikatu) okienko bufora odbioru musi zostać zwolnione przez „rozkaz zwolnienia bufora odbioru“, aby sterownik mógł do okienka przesunąć następny komunikat. Jeżeli rozkaz ten nie zostanie wydany, to ten sam komunikat będzie nieustannie przetwarzany powodując przepełnienie RXFIFO, ponieważ nie będą przesuwane następne odbierane ramki.
- Gdy częstość ramek jest duża, i wiele ramek jest wysyłanych jedna za drugą, a kolejne rozkazy mogą

być przesuwane niedostatecznie sprawnie, powstaje ryzyko szybkiego przepełnienia RXFIFO. Wobec takiego niebezpieczeństwa, trzeba użyć odpowiednio szybkiego mikrosterownika/komputera z wysokiej jakości oprogramowaniem. Przepełnienie RXFIFO jest sygnalizowane przez sterownik ustawieniem bitu błędu, czyli bitu przepełnienia danych w rejestrze stanu. Komunikat, który miał zostać przesunięty do RXFIFO (i który wywołał przepełnienie), zostaje wtedy skasowany i stracony.

EE

