

Interfejsy CAN i LIN – zastosowania, część 1

Więcej niż 23% kosztów wytworzenia współczesnego samochodu osobowego średniej i wyższej klasy stanowi elektronika. Mnogość urządzeń elektronicznych w samochodzie pociąga za sobą konieczność zapewnienia sprawnej transmisji danych, sterowania i zasilania do wszystkich modułów. Obecnie istnieją trzy sposoby rozwiązania tego problemu:

- wykorzystanie istniejącej dwuprzewodowej instalacji elektrycznej do przenoszenia informacji w postaci sygnałów zmodulowanych (około 100 kHz). Wadą tego rozwiązania jest istnienie wysokich wahań napięcia w instalacji elektrycznej (dla nominalnego napięcia 12 V występują przepięcia o amplitudzie do 40 V podczas włączania dużych obciążeń oraz krótkotrwałe skoki napięcia sięgające 100 V).
- wykorzystanie osobnych przewodów sterujących do każdego z modułów sterowanych. Wadą tego rozwiązania jest znaczne zwiększenie kosztów wytworzenia, masy i awaryjności instalacji.
- wykorzystanie specjalizowanego interfejsu sterującego. Jediną wadą tego rozwiązania jest konieczność zastosowania dodatkowych modułów sterujących pracą magistrali co przy cenach obecnych mikrokontrolerów wydaje się być sprawą marginalną.

W artykule skoncentrujemy się na opisanu obecnych specjalizowanych interfejsów komunikacyjnych wykorzystywanych w technice motoryzacyjnej. Idealne rozwiązanie powinno być systemem otwartym, odpornym na zakłócenia, wymagającym niskich zasobów sprzętowych do obsługi i posiadającym krótkie czasy transmisji. Dodatkowo powinno umożliwiać różne szybkości połączeń i powinno umożliwiać diagnostykę systemu.

Obecnie na rynku istnieją dwa najbardziej popularne sposoby transmisji:

- CAN (Controller Area Network),
- LIN (Local Interconnection Network).

W dalszej części artykułu przedstawione zostaną podstawowe informacje na temat obu protokołów wraz z przykładami aplikacji na bazie oferty firmy STMicroelectronics.

Kilometry przewodów i poziom komplikacji współczesnych instalacji samochodowych skłoniły producentów do „odchudzenia” samochodów poprzez wprowadzenie specjalizowanych interfejsów komunikacyjnych. Najpopularniejsze obecnie to CAN i LIN.

STMicroelectronics posiada w swojej ofercie podzespoły pozwalające stworzyć własne rozwiązania wykorzystujących oba interfejsy minimalnym nakładem środków i czasu.

Protokół CAN

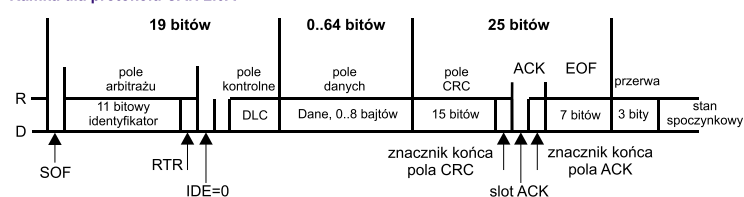
Protokół CAN, wynaleziony przez firmę Bosch w 1986 roku jest jednym z najbardziej rozpowszechnionych protokołów komunikacyjnych stosowanych w technice motoryzacyjnej. Umożliwia on osiągnięcie szybkości transmisji do 1 Mbd (dla sieci, których długość nie przekracza 40 m), co czyni go uży-

tecznym dla systemów pracujących w czasie rzeczywistym.

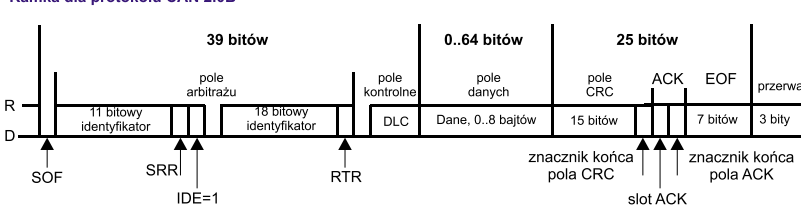
Podstawowe parametry protokołu

- magistrala szeregową asynchroniczną jedнопrzewodową (*low speed CAN* – ISO 11519) lub dwuprzewodową (*high speed CAN* – ISO 11898).
- dane transmitowane w kodzie NRZ

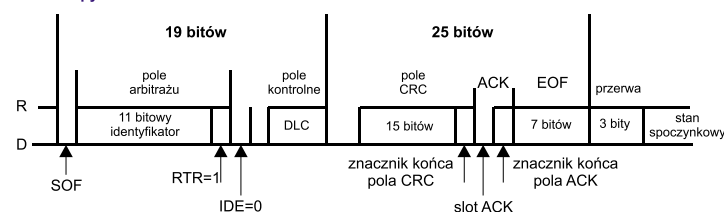
Ramka dla protokołu CAN 2.0A



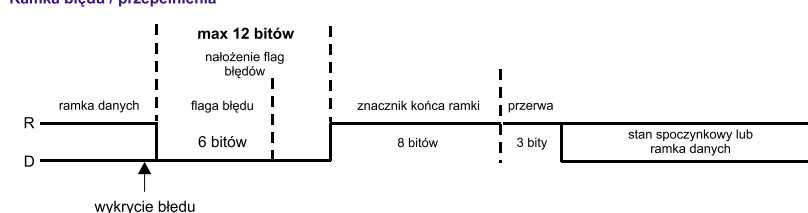
Ramka dla protokołu CAN 2.0B



Ramka zapytania



Ramka błędu / przepelnienia



Rys. 1.

Tab. 1. Znaczenie bitów pola DLC

MSB						LSB	
rezerwa	rezerwa	DLC3	DLC2	DLC1	DLC0	Liczba bajtów przesyłanych w bieżącej ramce	
X	X	D	D	D	D	0	
X	X	D	D	D	R	1	
X	X	D	D	R	D	2	
X	X	D	D	R	R	3	
X	X	D	R	D	D	4	
X	X	D	R	D	R	5	
X	X	D	R	R	D	6	
X	X	D	R	R	R	7	
X	X	R	D	D	D	8	

(*Non Return to Zero*). Poziomy logiczne nazywane są tu *dominant* (0) i *recessive* (1). W dalszej części niniejszego opracowania stosowane będą oznaczenia skrócone (odp. D i R).

- podłączenia do magistrali typu iloczyn na drucie (*wired-AND*), co powoduje że spoczynkowym stanem magistrali jest stan *recessive*, natomiast w celu zmiany stanu magistrali na *dominant* wystarczy zmiana stanu na linii w jednym z urządzeń.
- każde urządzenie dołączone do magistrali dosynchronizowuje się do zbczozy opadających transmitowanych sygnałów.
- w celu poprawy synchronizacji urządzeń na magistrali wprowadzono mechanizm polegający na wprowadzaniu bitu o przeciwnym znaku po wystąpieniu kilku (tu 5) bitów o tym samym poziomie (*stuffing bits*).
- zaimplementowano mechanizm CSMA/CA (*Carrier Sense Multiple Access/Collision Avoidance*).
- architektura typu multi-Master dopuszczające transmisję wielu modułów na magistrali.
- komunikacja na magistrali zorientowana obiektowo. Nie występuje tu pojęcie adresowania pakietów. Każda wiadomość przesyłana magistralą zawiera w sobie 11-bitowy (dla CAN 2.0B Passive) lub 29-bitowy (dla CAN 2.0B Active) identyfikator który określa transmitowane dane. Każdy z modułów podłączonych do magistrali monitoruje magistralę i wybiera pakiety dotyczące jego funkcjonowania.
- przesyłane informacje mają przydzielone priorytety.

Protokół definiuje też bardzo rozbudowane mechanizmy obsługi błędów transmisji:

- detekcja – każde urządzenie stale monitoruje magistralę. W przypadku wykrycia naruszenia protokołu, zgłaszany jest błąd.
- sygnalizacja – dowolne urządzenie podłączone do magistrali, po wykryciu błędu, natychmiast powiadamia swój moduł nadrzędny jednocześnie przery-

wając bieżącą transmisją. Nadawca przekłamaną informację ponawia transmisję po upływie określonego czasu.

- ograniczenie – każde urządzenie podłączone do magistrali posiada liczniki wykrytych błędów transmisji (zwiększanych przy wystąpieniu błędu transmisji i zmniejszanych przy poprawnej transmisji). W zależności od stanu wypełnienia tych liczników urządzenie może znajdować się w jednym z określonych stanów. W protokole CAN zdefiniowane są cztery różne rodzaje ramek:

- ramka danych (*data frame*) – transmitowanie danych do modułu(ów),
 - ramka błędu (*error frame*) – zgłoszenie wykrytego błędu na magistrali,
 - ramka zapytania (*remote frame*) – żądanie informacji od modułu,
 - sygnał przeładowania (*overload frame*) – żądanie opóźnienia transmisji.
- Na rys. 1 przedstawiono skład poszczególnych ramek z zaznaczeniem najważniejszych pól.

Funkcje poszczególnych pól są następujące:

- SOF (*Start of Frame*) – znacznik początku ramki (stan D na magistrali).
- RTR (*Remote Transmission Request*) – stan R na tej linii oznacza żądanie transmisji danych (ramki zapytania).
- IDE (*Identifier Extension Bit*) – stan D ustalany jest dla ramki w formacie standardowym, stan R ustalany dla ramki w formacie rozszerzonym.
- DLC (*Data Length Code*) – 6-bitowe pole określające liczbę bajtów przesyłanych w ramce. Dokładne znaczenie poszczególnych pól przedstawiono w tab. 1.
- Slot ACK – pole w którym nadawca ramki oczekuje na potwierdzenie odebrania ramki.
- EOF (*End of Frame*) – znacznik końca ramki (7 bitów typu R na magistrali).

Przebieg transmisji

W dużym uproszczeniu prawie każdy rodzaj ramki w standardzie CAN (wyjątkiem są ramka błędu i przepełnienia) można podzielić na trzy fazy:

- fazę arbitrażu, podczas której przesyłany jest identyfikator wiadomości,
- fazę transmisji danych, podczas której przesyłane są właściwe dane,

- fazę potwierdzenia, podczas której nadawca wiadomości oczekuje na potwierdzenie odbioru ramki (wymuszenie stanu R na magistrali) przez adresata (lub wszystkich adresatów).

Wszystkie urządzenia podłączone do magistrali mogą rozpocząć transmisję w dowolnym momencie pod warunkiem, że magistrala jest w stanie spoczynkowym (*Idle*). W przeciwnym przypadku należy czekać na zakończenie bieżącej transmisji. W momencie gdy kilka urządzeń jednocześnie rozpoczyna transmisję, urządzenia wysyłające bit D przejmują kontrolę nad magistralą zgodnie z regułą iloczynu na drucie. Proces ten realizowany jest w początkowej fazie transmisji (tzw. trakcie fazy arbitrażu, podczas transmisji identyfikator wiadomości). Na rys. 2 przedstawiono przykład arbitrażu trzech urządzeń (A, B i C) wygranego przez urządzenie A.

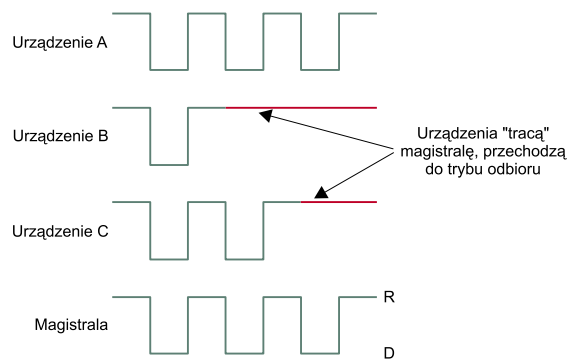
Po zakończeniu fazy arbitrażu, zaczyna się faza transmisji danych, podczas której nadaje tylko jedno urządzenie. W przypadku, gdy na tym etapie wykryje ono niezgodność pomiędzy stanem magistrali a wysyłanymi danymi, zgłaszany jest błąd (tj. Wysyłana jest ramka błędu).

Po zakończeniu fazy transmisji danych urządzenie (lub urządzenia) odbierające sprawdzają CRC transmitowanych danych i porównują otrzymany wynik z zawartym w ramce. W przypadku braku zgodności wysyłana jest ramka błędu. W przypadku uzyskania zgodności urządzenie odbierające powinny wystawić sygnał potwierdzenia transmisji (znak D). W przypadku braku tego znaku urządzenie nadawcze zgłasza sygnał błędu.

W przypadku stwierdzenia wystąpienia sygnału potwierdzenia na magistralę urządzenie nadawcze przesyła 7 kolejnych bitów typu R stanowiących znacznik końca transmisji.

Na podstawie materiałów firmy STMicroelectronics opracował

Artur Iwanicki



Rys. 2.