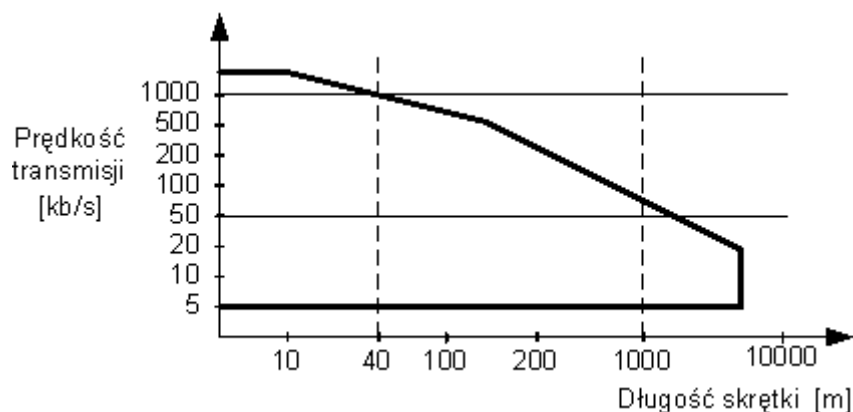


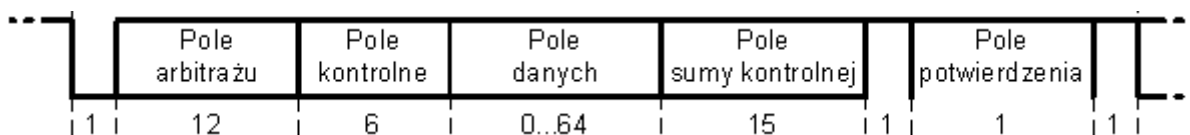
1.1.1. CAN

CAN (*Controller Area Network*) jest asynchronicznym interfejsem szeregowym, pracującym w trybie „Halfduplex” w konfiguracji „Master-Slave”. Interfejs został zaprojektowany do komunikacji między urządzeniami elektronicznymi w pojazdach samochodowych, z tego względu położono w nim duży nacisk na bezpieczeństwo (detekcja błędów). Dzięki swoim licznym zaletom CAN jest szeroko stosowany w przemyśle. Standard CAN określa zbiór zasad przeprowadzania transmisji danych między urządzeniami z wykorzystaniem jednej linii, nie narzuca jednak medium, za pomocą którego powinien przesyłany być sygnał. Jedynym wymogiem jest możliwość zaimplementowania stanu dominującego i recesywnego (rozdz. 3.3.2 o I²C). Najbardziej popularnym, a jednocześnie najtańszym, medium używanym do przesyłania w sieciach CAN jest kabel typu „skrętka”. Maksymalna prędkość transmisji w sieci CAN wynosi 1Mb/s. Oczywiście prędkość ta zmniejsza się wraz ze wzrostem odległości między urządzeniami (rys. 3.3.4.1).



Rys. 3.3.4.1. Wykres zależności maksymalnej prędkości transmisji od długości kabla typu „skrętka” w sieci CAN

Tak jak w przypadku interfejsu I²C standard CAN nie określa maksymalnej liczby urządzeń podłączonych do jednej linii (sieci). Każde z urządzeń może być zarówno nadajnikiem jak i odbiornikiem. W odróżnieniu od I²C nie ma tutaj urządzeń typu „Master/Slave”, co w praktyce oznacza, że każde z urządzeń może inicjować transmisję. (sposób unikania kolizji na łączu opisany zostanie w dalszej części rozdziału). Ramkę komunikatu w interfejsie CAN pokazano na rysunku 3.3.4.2.

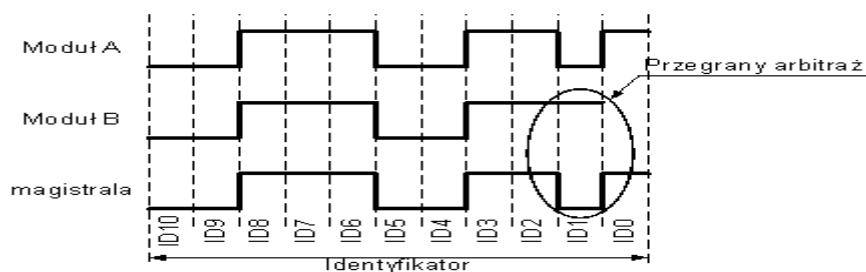


Rys. 3.3.4.2. Ramka transmisji danych interfejsu CAN

Można w niej wyróżnić cztery rodzaje bitów, są to: bity synchronizacji, bity kontrolne, bity arbitrażu, bity danych i bity sumy kontrolnej CRC. (Cyclic Redundant Code), bity

potwierdzenia. **Bity synchronizacji** pozwalają odbiornikowi monitorować fazę sygnału zegarowego nadajnika (częstotliwość zegara, tożsama z prędkością transmisji jest z góry ustalona w całej sieci). Tworzące pole arbitrażu **bity identyfikacji** nie są adresem urządzenia odbierającego dane, jak to mam miejsce w interfejsie I²C, ale identyfikatorem danych przesyłanych za pomocą komunikatu. Stosowana jest tutaj komunikacja typu „**Broadcast**”. Każde urządzenie podłączone do sieci filtruje przychodzące komunikaty i odbiera tylko te, które zawierają określony identyfikator (bity identyfikacji). Długość pola **bitów danych** jest zmienna i może zawierać od zera do ośmiu bajtów (ilość bajtów danych określają 4 bity w polu DLC (*Data Length Code*)). Przesyłany w polu sumy kontrolnej **kod CRC** jest obliczany na podstawie danych zawartych w komunikacie od bitu startu do końca pola danych. Jest on obliczany zarówno przez nadajnik jak i przez odbiornik. Jeżeli odbiornik wykryje różnicę w CRC obliczonym i odebranych traktuje odebrane dane jako błędne.

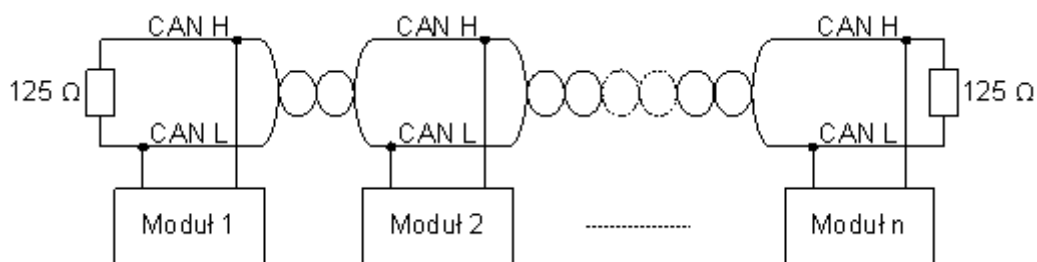
Konsekwencją braku rozróżnienia urządzeń podłączonych do sieci CAN na urządzenia typu „Master” i „Slave” była konieczność zdefiniowania mechanizmu unikania kolizji na łączu. Zdecydowano się na unikanie kolizji poprzez przyporządkowanie każdemu komunikatowi priorytetu. Oznacza to, że gdy wiele urządzeń chce wysłać komunikat, kontrole nad łączem przejmuje urządzenie, którego komunikat ma najwyższy priorytet. Priorytet komunikatu stanowi zawartość jego pola arbitrażu (bity identyfikacji). Wyższy priorytet mają komunikaty, których słowo utworzone przez bity identyfikatora ma niższą wartość. Przykład procedury arbitrażu pokazano na rys. 3.3.4.3. Procedura wykorzystuje istnienie stanów recesywnego („1”) i dominującego („0”) zdefiniowanych w rozdziale 3.3.2. Urządzenia, które chcą uzyskać dostęp do magistrali wystawiają po kolei na magistralę poszczególne bity identyfikacji (od najstarszego do najmłodszego). Jeżeli któreś z urządzeń zauważy, że wystawiło stan „1” a na magistrali znajduje się stan „0” przestaje nadawać ponieważ oznacza to, że na magistrali nadaje urządzenie z wyższym priorytetem.



Rys. 3.3.4.3. Przykład procedury arbitrażu dwóch nadajników.

Moduł B traci dostęp do magistrali ponieważ wartość jego pola arbitrażu jest większa od wartość pola arbitrażu modułu A.

Jak sama nazwa wskazuje interfejs CAN (*Controller Area Network*) został stworzony do komunikacji między kontrolerami w tym mikrokontrolerami. Obecnie wszyscy ważniejsi producenci mikrokontrolerów oferują modele zawierające w sobie kontrolery CAN. Najbardziej popularnym sposobem podłączenia mikrokontrolerów w sieć CAN jest zastosowanie linii różnicowej za pomocą kabla typu „skrętka” (rys. 3.3.4.4). W celu zapewnienia dopasowania elektrycznego wejść/wyjść mikrokontrolerów z linią różnicową CAN powszechnie stosuje się specjalizowane konwertery TTL/CAN.



Rys. 3.3.4.4. Przykład podłączenia trzech kontrolerów do sieci CAN z łączem transmisyjnym w postaci „skrętki”