



Politechnika Wroclawska

**Komputerowe sieci miejscowe
(sieci polowe, sieci przemysłowe, wymiana danych w
systemach sterowania)**

Wykład 2

**Protokoły transmisji
(opracowano na podstawie materiałów ze strony
<http://wazniak.mimuw.edu.pl/>)**

Dr inż. Zbigniew Zajda
Instytut Informatyki, Automatyki i Robotyki
Wydział Elektroniki Politechniki Wrocławskiej





Program wykładu

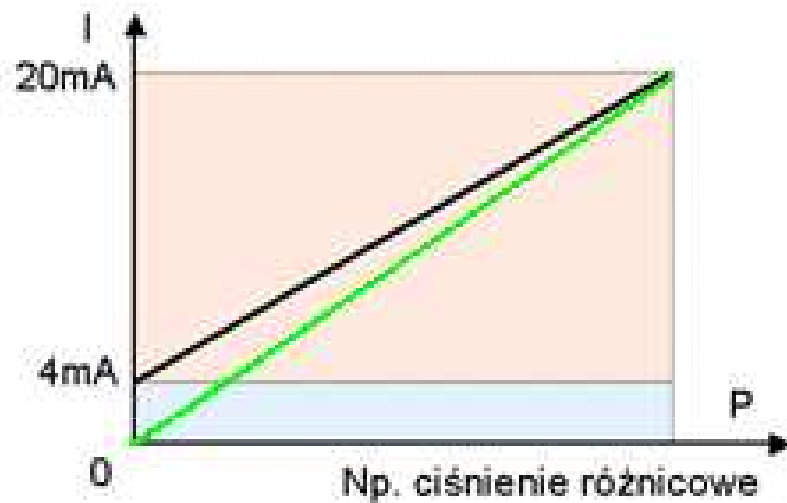
Protokoły transmisji

Protokoły transmisji w sieciach przemysłowych

- Modbus
- Hart
- AS-I
- CAN

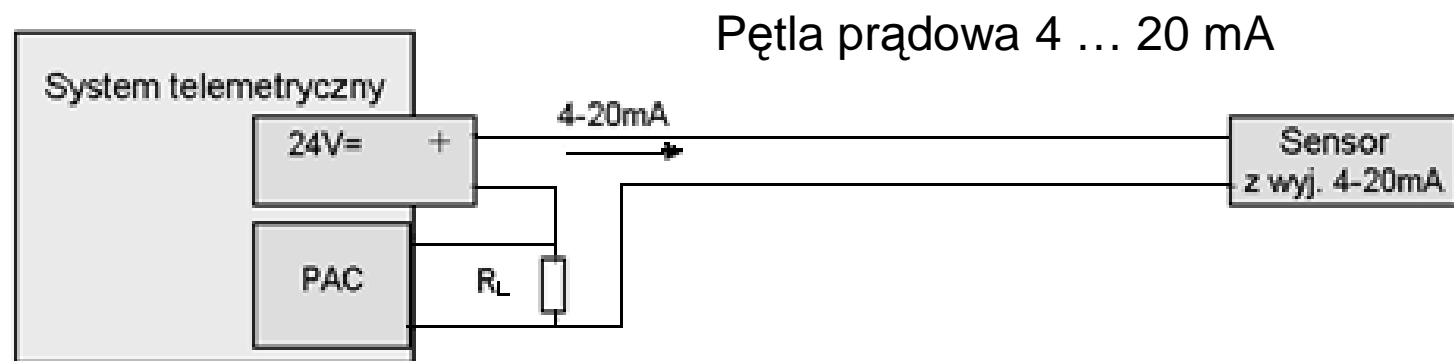


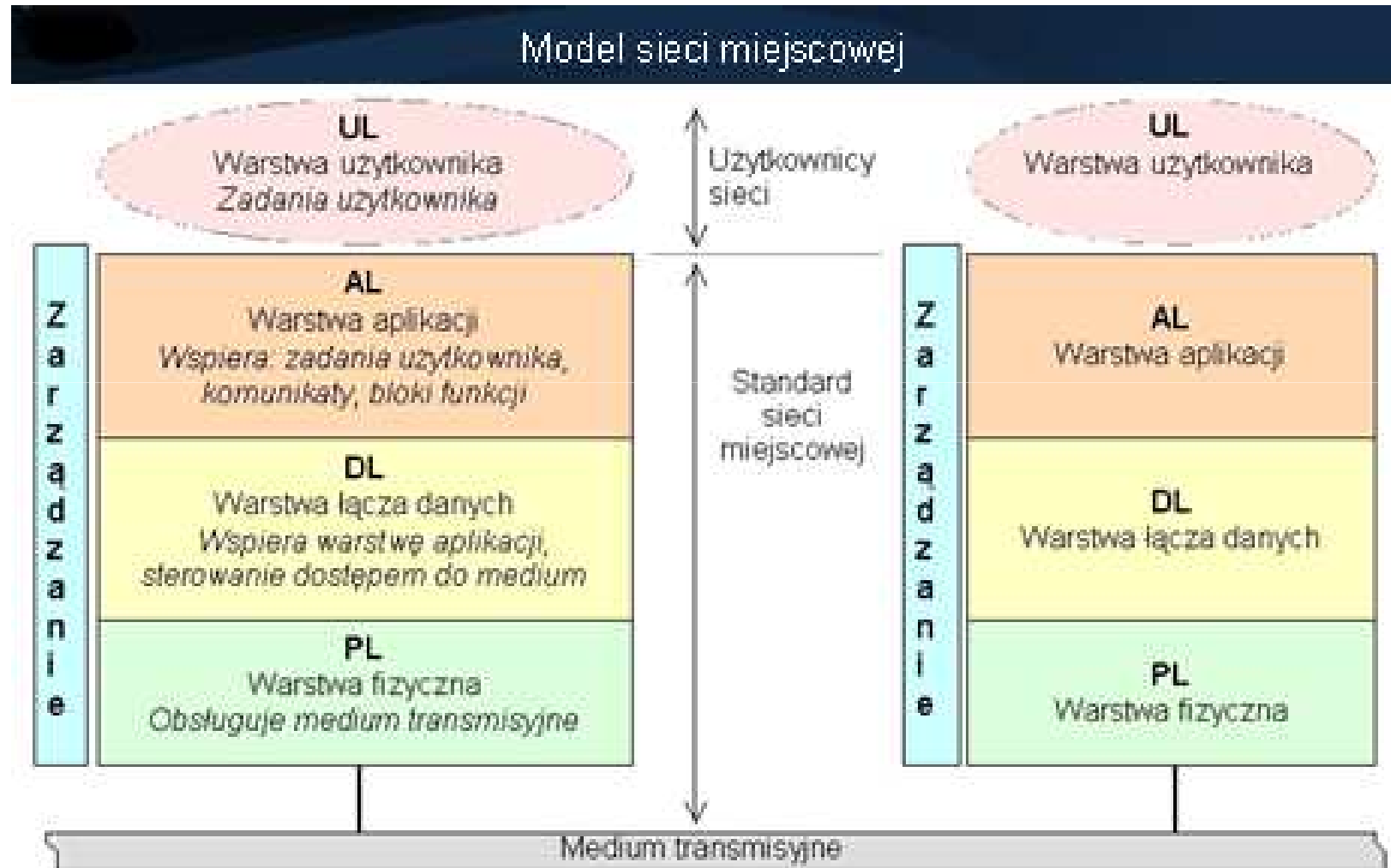
Protokoły transmisji - Standardy analogowe



Sygnał prądowy:

- 0 ... 20 mA
- 420 mA







Protokoły transmisji- MODBUS

Protokół Modbus został opracowany w firmie Modicon w 1980r. Jest standardem wykorzystywanym przez wielu producentów urządzeń sterujących i pomiarowych wyposażonych w interfejsy szeregowo: RS-232, RS-422 lub RS-485. Większość programowych pakietów narzędziowych SCADA (sterowanie nadrzędne i akwizycja danych) wyposażona jest w procedury komunikacyjne realizujące ten protokół .

O popularności protokołu Modbus zdecydował moment pojawienia się na rynku, prostota zastosowanych w nim rozwiązań, jawność specyfikacji protokołu oraz dostęp do magistrali w trybie master - slave, zabezpieczenie komunikatów przed błędami transmisji, potwierdzenie wykonania rozkazów i sygnalizacja błędów oraz mechanizmy zapobiegające zawieszaniu się systemu. Cechy te pozwalają na łatwą i tanią implementację w dowolnym urządzeniu posiadającym mikrokontroler .

W modelu ISO/OSI protokół Modbus wykorzystuje trzy warstwy:
1 - fizyczną, 2 - łącza danych oraz 7-aplikacji.



Protokoły transmisji- MODBUS

Najczęściej stosowany w automatyce przemysłowej jest protokół Modbus współpracujący z interfejsem RS485, gdzie występuje jedno urządzenie nadrzędne (master) inicjalizujące transakcje (wysyłające polecenie), natomiast pozostałe urządzenia podrzędne (slaves), wykonują polecenia mastera i odsyłają odpowiedź.

W danej chwili tylko jeden slave może odpowiadać na zapytanie mastera. Nie ma możliwości komunikacji pomiędzy urządzeniami podrzędnymi. Typowym masterem sieci jest urządzenie z procesorem głównym (host procesor), zawierające programowalny panel na przykład komputer PC lub nadrzędny sterownik logiczny, a typowy slave to programowalny sterownik logiczny.

Węzły podrzędne (slaves) są wykorzystywane do sterowania oraz zbierania danych z urządzeń peryferyjnych takich jak: mierniki, liczniki, przetworniki A/C i C/A, czujniki, przekaźniki, sygnalizatory itp.



Protokoły transmisji- MODBUS

Najważniejsze cechy protokołu Modbus :

- zasada dostępu do łącza „query - response” („master-slave”) gwarantująca bezkonfliktowe współdzielenie magistrali przez wiele węzłów;
- węzeł nadrzędny (master) - komputer np. klasy PC steruje pracą sieci;
- węzły podrzędne (slaves) - pozostałe węzły nie podejmują samodzielnie transmisji, odpowiadają jedynie na zdalne polecenia od węzła nadrzędnego;
- każdy z węzłów podrzędnych posiada przypisany unikalny adres;
- dwa różne tryby transmisji ASCII (znakowy) lub RTU (binarny),
- komunikaty zawierające polecenia i odpowiedzi mają identyczną strukturę;
- maksymalna długość komunikatów wynosi 256 bajtów;
- znaki są przesyłane szeregowo od najmłodszego do najstarszego bitu.



Protokoły transmisji- MODBUS

Główne zalety protokołu Modbus :

- prostota zastosowanych w nim rozwiązań;
- jawność specyfikacji protokołu;
- zabezpieczenie przesyłanych komunikatów przed błędami;
- potwierdzanie wykonania rozkazów zdalnych i sygnalizacja błędów;
- stały format ramki i zestaw standardowych funkcji służących wymianie danych;
- mechanizmy zabezpieczające przed zawieszeniem systemu.



Protokoły transmisji- MODBUS

Ramka protokołu Modbus określa format przesyłanych wiadomości i zawiera: adres odbiorcy, kod funkcji reprezentujący żądane polecenie, dane dotyczące funkcji oraz słowo kontrolne zabezpieczające przesyłaną wiadomość.

Postać ramki zapytania wysyłanego przez jednostkę Master i ramki odpowiedzi jednostki slave jest podobna. Różnica polega na tym, że w polu danych ramki odpowiedzi występują dane, których dostarczenia żądała stacja master.



Protokoły transmisji- MODBUS



Ramka protokołu Modbus



Protokoły transmisji- MODBUS

Opis poszczególnych pól ramki:

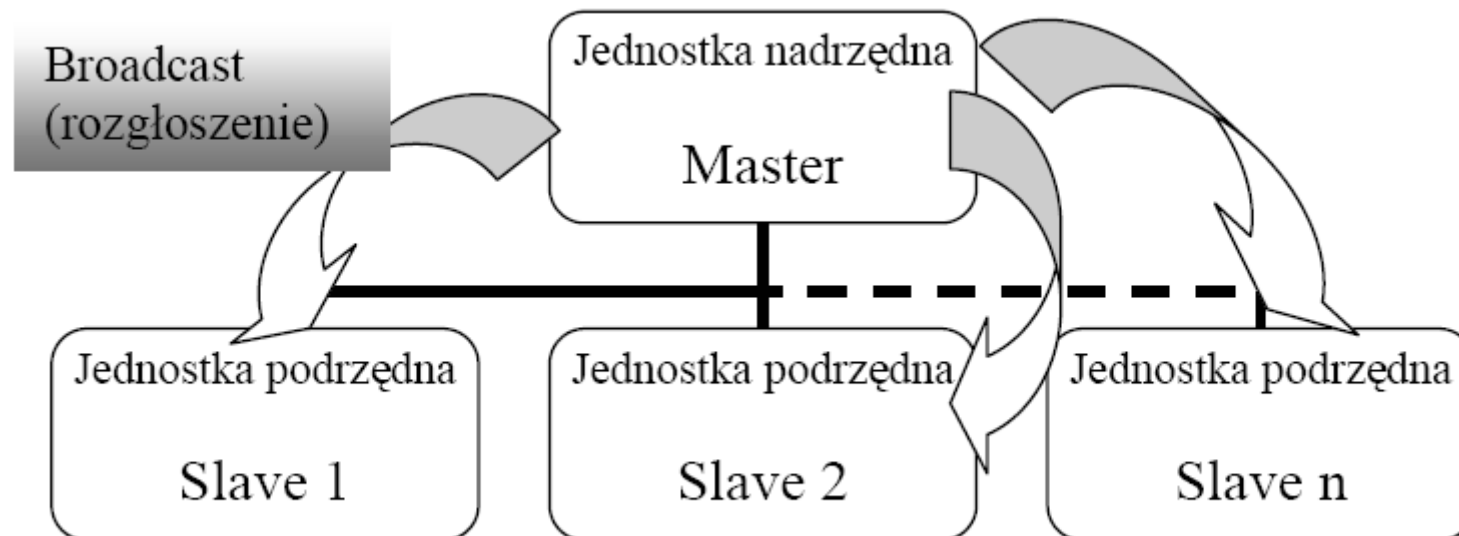
- *adres SLAVE*: liczba z zakresu 1 - 247, 0 - adres rozgłoszeniowy;
- *funkcja*: liczba z zakresu 1...127;
- *pole danych*: długość zależy od rodzaju wiadomości i może zawierać:
 - w przypadku zapytania - argumenty funkcji;
 - w przypadku pozytywnej odpowiedzi - argumenty funkcji;
 - w przypadku szczególnej odpowiedzi - kod błędu;
 - w niektórych przypadkach może być równa 0;
- *suma kontrolna*: wyznaczana z zawartości przesyłanego komunikatu.



Protokoły transmisji- MODBUS

Możliwe są dwa typy transakcji:

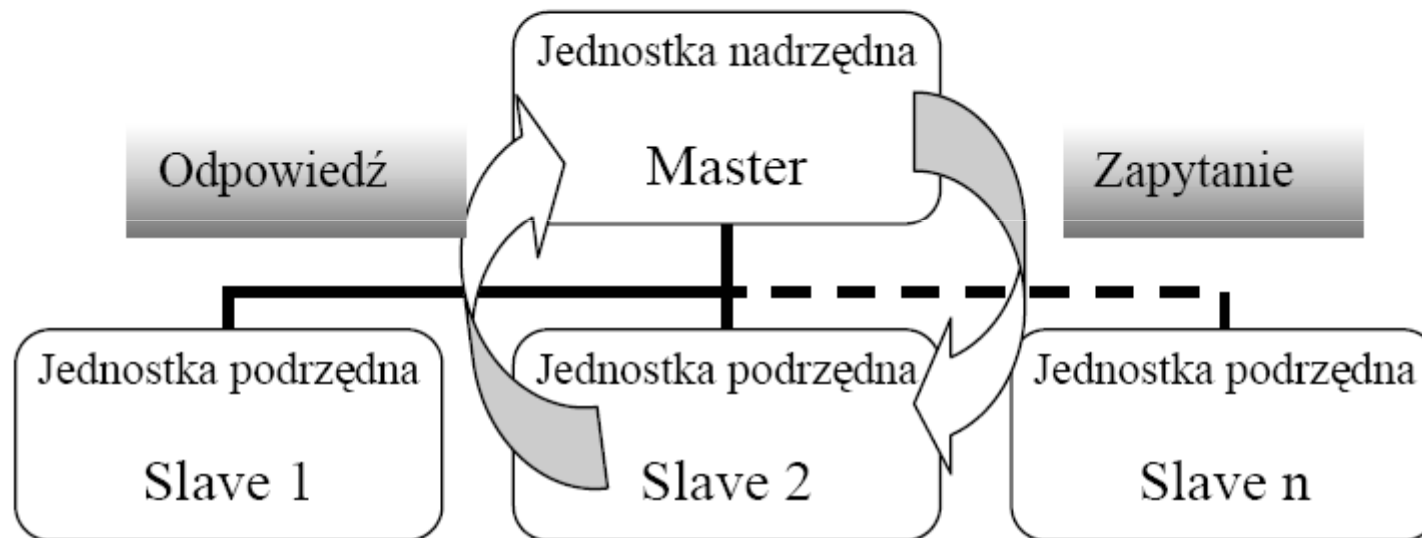
- „rozgłoszenie” (*Broadcast*) - jednostka master wysyła jedynie ramkę rozgłoszenia o adresie 0, który nie jest przypisany do konkretnego urządzenia slave, ale wszystkie urządzenia ją odbierają i wykonują zawartą w niej funkcję nie odsyłając odpowiedzi.





Protokoły transmisji- MODBUS

- „zapytanie - odpowiedź” (Query-Response) - jednostka master wysyła zapytanie i oczekuje na odpowiedź od wybranego urządzenia slave;





Protokoły transmisji- MODBUS

Typowe zachowanie urządzenia Slave po odbiorze zapytania i zweryfikowaniu adresu może wyglądać następująco:

- w przypadku poprawnego odbioru i bezbłędnej interpretacji ramki zapytania wysyłanej przez węzeł master odpowiada również zgodnie z formatem zdefiniowanym w protokole Modbus;
- gdy wystąpi błąd przy odbiorze wiadomości lub slave nie jest w stanie wykonać polecenia wysyła tzw. szczególną odpowiedź (*Exception Response*), w ramce odpowiedzi wysyłanej przez stację slave w polu kodu funkcji ustawiany jest najstarszy bit, natomiast w polu danych umieszczany jest kod błędu, umożliwiający węzłowi nadrzdnemu określenie przyczyny jego wystąpienia;
- w przypadku błędu sumy kontrolnej lub innych błędów w ramce zapytania nie zdefiniowanych w kodach błędów, slave nie wysyła odpowiedzi.



Protokoły transmisji- MODBUS

Standardowe kody błędów odpowiedzi wyjątkowej (szczególnej):

- 01 - niedozwolona funkcja
- 02 - niedozwolony zakres danych(adres)
- 03 - niedozwolona wartość danej
- 04 - błąd urządzenia Slave
- 05 - potwierdzenie pozytywne
- 06 - brak gotowości urządzenia Slave
- 07 - potwierdzenie negatywne
- 08 - błąd parzystości pamięci



Protokoły transmisji- MODBUS

Tryby transmisji

Protokół Modbus definiuje dwa tryby transmisji: ASCII (znakowy), RTU (binarny).

- Tryb ASCII:

każdy bajt w wiadomości przesyłany jest w postaci dwóch znaków ASCII,

ZNACZNIK POCZĄTKU	ADRES	FUNKCJA	DANE	KONTROLA LRC	ZNACZNIK KOŃCA
1 ZNAK ' : '	2 ZNAKI	2 ZNAKI	n ZNAKÓW	2 ZNAKI	2 ZNAKI 'CR' i 'LF'



Protokoły transmisji- MODBUS

Tryb RTU :

- wiadomości rozpoczynają się odstępem czasowym trwającym minimum $3,5 * TZ$, gdzie TZ - czas trwania pojedynczego znaku;
- cała ramka musi zostać przestana w postaci ciągłej tzn. odstęp pomiędzy kolejnymi znakami nie może być większy od $1,5 * TZ$.

ZNACZNIK POCZĄTKU	ADRES	FUNKCJA	DANE	KONTROLA CRC	ZNACZNIK KOŃCA
ODSTĘP 3,5 x	8 BITÓW	8 BITÓW	n x 8 BITÓW	16 BITÓW	PRZERWA 3,5 x



Protokoły transmisji- MODBUS

W węźle nadrzędnym ustalany jest pewien maksymalny czas oczekiwania na odbiór ramki zawierającej komunikat z odpowiedzią (Timeout). Jego wartość musi być tak dobrana, aby nawet najwolniejszy z węzłów podrzędnych zdążył odesłać odpowiedź. Przekroczenie tego czasu powoduje przerwanie transakcji. Przyczyn braku odpowiedzi może być kilka.

Najczęstszymi przyczynami braku odpowiedzi są:

- wystąpienie błędu transmisji ramki polecenia,
- zaadresowanie nieistniejącego węzła podrzędnego.



Protokoły transmisji- MODBUS

Format znaku przy transmisji szeregowej

W standardzie Modbus znaki są przesyłane szeregowo od najmłodszego do najstarszego bitu.

Organizacja jednostki informacyjnej w trybie ASCII:

- 1 bit startu,
- 7 bitów pola danych, jako pierwszy wysyłany jest najmniej znaczący bit,
- 1 bit kontroli parzystości (nieparzystości) lub brak bitu kontroli parzystości,
- 1 bit stopu przy kontroli parzystości lub 2 bity stopu przy braku kontroli parzystości.

START	1	2	3	4	5	6	7	PAR	STOP
START	1	2	3	4	5	6	7	STOP	STOP



Protokoły transmisji- MODBUS

Organizacja jednostki informacyjnej w trybie RTU:

- 1 bit startu,
- 8 bitów pola danych, jako pierwszy wysyłany jest najmniej znaczący bit,
- 1 bit kontroli parzystości (nieparzystości) lub brak bitu kontroli parzystości,
- 1 bit stopu przy kontroli parzystości lub 2 bity stopu przy braku kontroli parzystości.

START	1	2	3	4	5	6	7	8	PAR	STOP
START	1	2	3	4	5	6	7	8	STOP	STOP



Protokoły transmisji- MODBUS

Standardowe funkcje Modbus:

Funkcje protokołu Modbus dzielą się na funkcje publiczne (standardowe) i funkcje definiowane przez użytkownika. Są to rozkazy wysyłane przez jednostki nadrzędne oraz interpretowane i wykonywane przez jednostki podrzędne (Slaves).

Większość opisanych w specyfikacji protokołu poleceń służy do realizacji operacji na sprzętowych zasobach sterowników przemysłowych - cyfrowych wejściach (INPUTS) i wyjściach (COIL) oraz rejestrach wejściowych (INPUT REGISTERS) oraz rejestrach wyjściowych (HOLDING REGISTERS). Rejestry są 16-bitowe i mogą mapować np. układy liczników lub przetworników A/C i C/A modułu wykonawczego.

Producenci sterowników przemysłowych wyposażonych w kontrolery przeznaczone dla standardu Modbus opracowują dla wykonywanych przez siebie urządzeń wiele własnych funkcji.



Protokoły transmisji- MODBUS

Typowe funkcje zdefiniowane w protokole Modbus:

Read Coil Status - odczyt bieżącego stanu grupy wyjść cyfrowych,

Read Input Status - odczyt stanu grupy wejść cyfrowych,

Read Holding Register - odczyt zawartości grupy rejestrów wyjściowych,

Read Exception Status - odczyt statusu urządzenia Slave,

Read Input Register - odczyt zawartości grupy rejestrów wejściowych

Diagnostics - test diagnostyczny,

Force Multiple Coils - ustawienie stanu grupy wyjść cyfrowych,

Preset Multiple Register - zapis do grupy rejestrów wyjściowych,

Report Slave ID - odczyt ID jednostki Slave,

Reset Communication Link - resetowanie połączenia,

Read General Reference - odczyt rejestrów w pamięci rozszerzonej,

Write General Reference - zapis do rejestrów w pamięci rozszerzonej,

Mask Write 4X Register - maskowanie grupy 4 rejestrów,

Read/Write 4X Register - odczyt/zapis grupy 4 rejestrów,

Read FIFO Queue - odczyt kolejki FIFO.



Protokoły transmisji- MODBUS

W protokole Modbus jako zabezpieczenie ramki wiadomości stosuje się sumę kontrolną. Jej wartość wyznacza urządzenie nadające dla zawartości przesyłanego komunikatu i umieszcza w ramce po części informacyjnej. Węzeł odbiorczy oblicza sumę kontrolną dla odebranego komunikatu i porównuje jej wartość z wartością otrzymaną. Niezgodność sum świadczy o wystąpieniu błędu. Dla trybu ASCII stosuje się sumę kontrolną typu LRC (Longitudinal Redundancy Check) natomiast dla trybu RTU - CRC (Cyclical Redundancy Check).

Generacja LRC

Wartość LRC (8-bitowa) jest dołączana na końcu ramki w postaci dwóch znaków ASCII. Obliczanie LRC polega na sumowaniu kolejnych 8-bitowych bajtów wiadomości, odrzuceniu przeniesień, a na koniec wyznaczeniu uzupełnienia dwójkowego wyniku. Sumowanie obejmuje całą wiadomość za wyjątkiem znaczników początku i końca ramki.



Protokoły transmisji- MODBUS

Generacja CRC

Słowo kontrolne CRC to 16-bitowa wartość dołączana do ramki w postaci dwóch 8-bitowych znaków.

Obliczanie CRC realizowane jest według następującego algorytmu:

- 1) załadowanie wartości FFFFh do 16-bitowego rejestru;
- 2) pobranie bajtu z bloku danych (zabezpieczana wiadomość) i wykonanie operacji EXOR z młodszym bajtem rejestru, umieszczenie rezultatu w rejestrze;
- 3) przesunięcie zawartości rejestru w prawo o jeden bit połączone z wpisaniem 0 na najbardziej znaczący bit (MSB=0);
- 4) sprawdzenie stanu najmłodszego bitu (LSB) w rejestrze, jeżeli jego stan równa się 0, to następuje powrót do kroku 3 (kolejne przesunięcie), jeżeli 1, to wykonywana jest operacja EXOR rejestru ze stałą A001h;
- 5) powtórzenie kroków 3 i 4 osiem razy, co odpowiada przetworzeniu całego bajtu;
- 6) powtórzenie sekwencji 2, 3, 4, 5 dla kolejnego bajtu wiadomości, kontynuacja tego procesu aż do przetworzenia wszystkich bajtów wiadomości;
- 7) zawartość rejestru po wykonaniu wymienionych operacji jest poszukiwaną wartością CRC.



Protokoły transmisji- MODBUS

a) POLECENIE			
Nazwa pola	Przykład (hex)	TRYB ASCII (znaki)	TRYB RTU (8-bitowe pole)
Znacznik początku		:	brak znacznika
Adres slave	07	0 7	0000 0111
Kod funkcji	04	0 4	0000 0100
Adres początkowy HI	00	0 0	0000 0000
Adres początkowy LO	03	0 3	0000 0011
Ilość rejestrów HI	00	0 0	0000 0000
Ilość rejestrów LO	02	0 2	0000 0010
Słowo kontrolne		LRC (2 znak)	CRC (16 bitów)
Znacznik końca		CR LF	brak znacznika
Całkowita ilość bajtów:		17	8
b) ODPOWIEDŹ			
Nazwa pola	Przykład (hex)	TRYB ASCII (znaki)	TRYB RTU (8-bitowe pole)
Znacznik początku		:	brak znacznika
Adres slave	07	0 7	0000 0111
Kod funkcji	04	0 4	0000 0100
Ilość bajtów	04	0 4	0000 0100
Dane: rejestr 1 HI	08	0 8	0000 1000
Dane: rejestr 1 LO	01	0 1	1100 0001
Dane: rejestr 2 HI	5A	5 A	0101 1010
Dane: rejestr 2 LO	3E	3 E	0011 1110
Słowo kontrolne		LRC (2 znak)	CRC (16 bitów)
Znacznik końca		CR LF	brak znacznika
Całkowita ilość bajtów:		19	9

Realizacja zdalnego odczytu grupy rejestrów



Protokół komunikacyjny HART? (Highway Addressable Remote Transducer) opracowano w firmie Rosemount w roku 1986. Stanowi on zgodnie z zamysłem twórców naturalne rozszerzenie przemysłowego standardu analogowego 4-20mA o cyfrową komunikację z „inteligentnymi” (mikroprocesorowymi) układami wejścia/wyjścia. Wspomniany standard ma już długą historię oraz bardzo dużo aplikacji w przemyśle, głównie chemicznym i pomimo rozwoju technik cyfrowych pojawia się również w nowych instalacjach.

Architekturę różnych sieci najczęściej porównuje się z modelem odniesienia OSI. Model ten nie uwzględnia specyfiki sieci miejscowych, które z założenia są sieciami czasu rzeczywistego, a ponieważ przeznaczone są do pracy w warunkach przemysłowych muszą charakteryzować się zwiększoną odpornością na zakłócenia. Różnice dotyczą też fizycznych środków łączności i topologii sieci. Przy uwzględnieniu tych różnic model OSI stanowi użyteczną bazę odniesienia. Sieć HART stosuje tylko trzy warstwy, 1, 2 i 7.



Format ramki

Model OSI

7	Warstwa aplikacji
6	Warstwa prezentacji
5	Warstwa sesji
4	Warstwa transportowa
3	Warstwa sieciowa
2	Warstwa łącza danych
1	Warstwa fizyczna

Sieć miejscowa HART

Instrukcje
Reguły protokołu HART
Bell 202



Warstwa fizyczna dotyczy technicznych aspektów przesyłania danych w postaci strumienia bitów. HART wykorzystuje modulację FSK (ang. Frequency Shift Keying). Przyjęto standard Bell 202. Jest to metoda kluczenia częstotliwości z fazą ciągłą: jeden cykl 1200Hz reprezentuje binarne 1, natomiast dwa cykle 2200Hz - binarne 0. Taki dobór częstotliwości pozwala na uniknięcie, tzw. zniekształceń włączeniowych.

Standard Bell 202 opracowano w USA dla potrzeb transmisji cyfrowej w sieci telefonicznej. Powstało szereg modemów scalonych dla tego standardu. Najbardziej popularne to SYM20C15 z firmy Symbios Logic, HT20C12 z SMAR Research Corp. oraz FX614 z CML Semiconductor Products. Przyjęcie takiego rozwiązania pozwala na bezproblemowe przekazywanie ramek HART poprzez sieć telefoniczną.

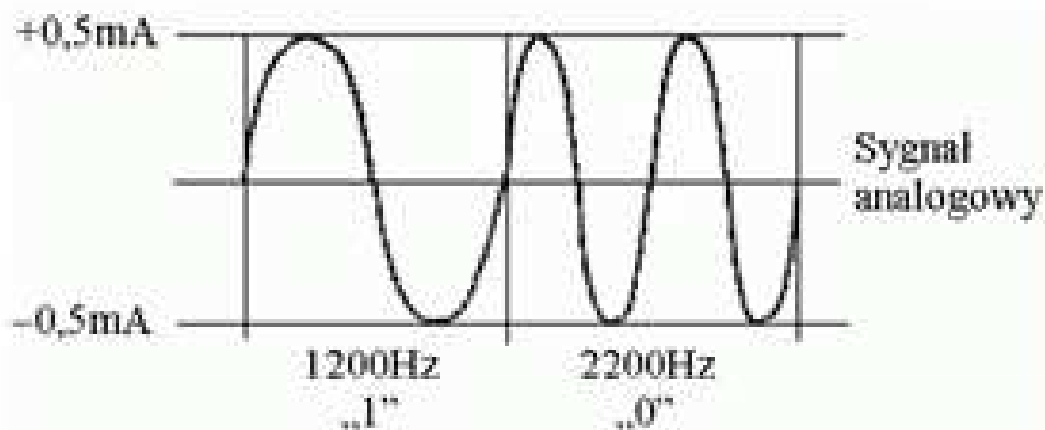
Sieć HART może pracować w konfiguracji „punkt-punkt” (pojedynczy układ slave o adresie 0) albo „wielopunkt” zrealizowany na magistrali dwu- lub trzyprzewodowej. Pierwszy sposób stosowany jest wspólnie z transmisją analogową 4-20mA. Pasmo sygnału analogowego ograniczone jest do 25Hz. W drugim przypadku, w którym dopuszczona jest tylko transmisja cyfrowa, do sieci może być dołączonych maksymalnie 15 modułów. Każdy z nich musi mieć inny adres; dla pierwotnego master'a zarezerwowany jest adres 1, a dla wtórnego - 0. Moduły slave przełączone zostają do trybu stałego obciążenia prądem 4mA (instrukcja #6).



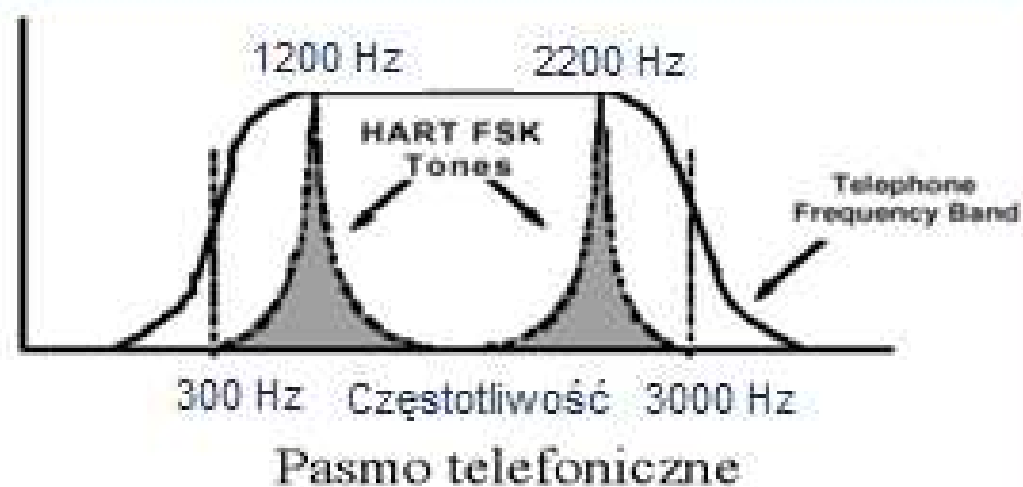
Protokoły transmisji



HART (ang. Highway Addressable Remote Transducer)



Modulacja w sieci HART



Widmo sygnału HART



Strumień bitów komunikatu HART dzielony jest na bajty (8 bitowe znaki), z których każdy jest kodowany zgodnie z regułami stosowanymi w układach UART: 1 bit startu, 8 bitów danych, 1 bit parzystości i 1 bit stopu. Komunikat rozpoczyna się wstępem, którego długość wynosi 5 lub 20 bajtów (przy braku synchronizacji); same znaki FFH. Specyfikacja HART przewiduje dwa rodzaje formatu ramki: krótki i długi. Formaty różnią się polem adresu. Znak startu (pole SC) służy do rozróżnienia kierunku transmisji (M ? S lub S ? M), zaznaczenia czy ramka została wygenerowana w trybie cyklicznego generowania odpowiedzi przez moduł slave (ang. burst) oraz określenia formatu ramki. Adres dla formatu krótkiego ma długość 1 bajtu. Najstarszy bit jest adresem master'a („1” - pierwotny, „0” - wtórny). Kolejny bit BM służy do przełączania slave z/do trybu „burst”. Następne dwa bity są zerami, a cztery ostatnie to pole adresu slave. Adres ramki długiej definiuje dwa pierwsze bity jak wyżej. Pozostałe sześć bitów oznacza producenta. Drugi bajt określa typ urządzenia. Dalsze sześć bajtów to unikatowy numer modułu slave. Pole statusu (2 bajty) występuje opcjonalnie w odpowiedzi modułu slave i dotyczy jego stanu. Pole danych może zawierać 0 - 25 bajtów, a treść jest zdefiniowana w polu instrukcji.



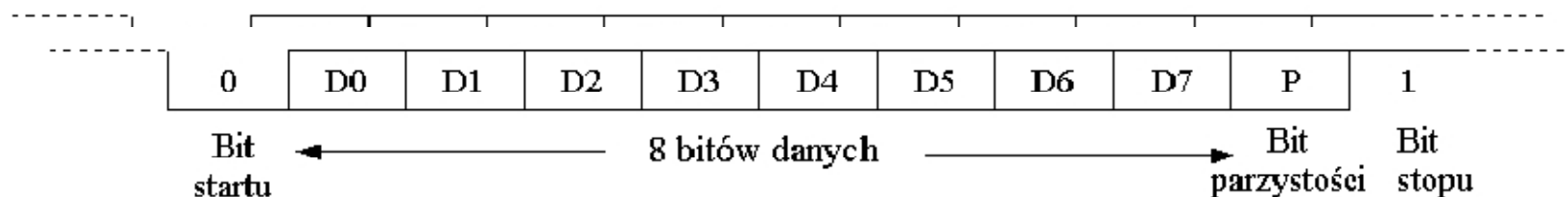
Protokół HART może być użyty w różnych trybach komunikacyjnych. Przewidziano możliwość stosowania dwóch modułów master (pierwotny i wtórny). Wtórny master w postaci podręcznego komunikatora może być użyty bez zakłócania komunikacji z masterem pierwotnym, służy do monitorowania i konfigurowania pracy systemu. Oba moduły master mają różne adresy, co pozwala na identyfikację odpowiedzi modułu (modułów) slave na rozkazy każdego z nich. Najczęściej stosowanym trybem jest komunikacja cyfrowa master-slave, realizowana jednocześnie z transmisją analogową 4-20mA. Odpowiada to konfiguracji sieciowej „punkt-punkt”. W tym przypadku można wyróżnić dwa węzły analogowe i dwa węzły cyfrowe. Protokół HART może być też użyty w konfiguracji „wielopunkt” zrealizowany na magistrali, ale wtedy mogą występować wyłącznie węzły cyfrowe. Opcjonalnym trybem komunikacyjnym jest „burst”, w którym pojedynczy moduł slave cyklicznie wysyła standardowe komunikaty. Szczelina czasowa między komunikatami umożliwia modułowi master zmianę rozkazu lub trybu.



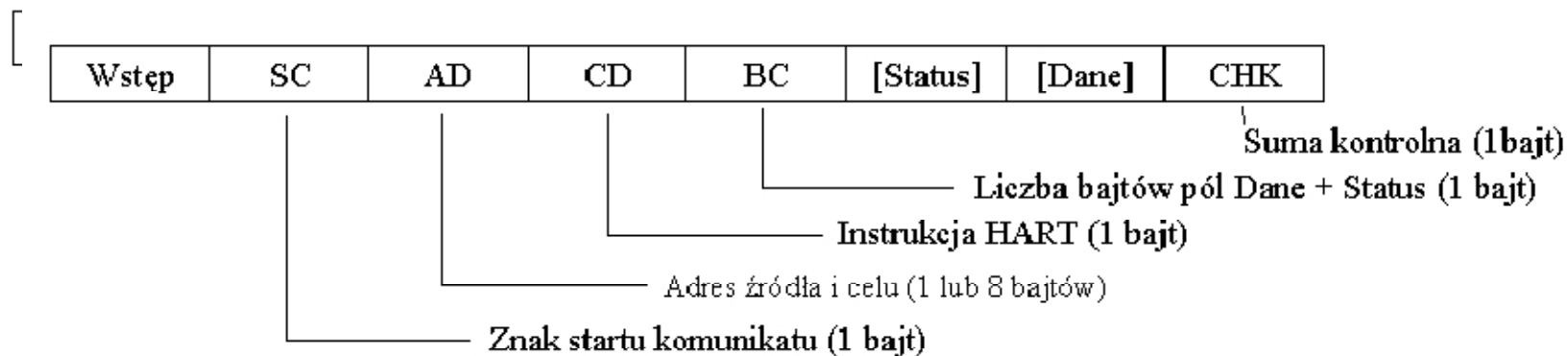
Protokoły transmisji



HART(ang. Highway Addressable Remote Transducer)



Format znaku



Struktura komunikatu HART



Politechnika Wroclawska

Protokoły transmisji



HART(ang. Highway Addressable Remote Transducer)



Model OSI

7	Warstwa aplikacji
6	Warstwa prezentacji
5	Warstwa sesji
4	Warstwa transportowa
3	Warstwa sieciowa
2	Warstwa łącza danych
1	Warstwa fizyczna

Sieć miejscowa AS-i

Instrukcje
Reguły protokołu AS-i
Modulacja PE, APM



Sieć ASi należy do najprostszych sieci miejscowych – łączy punkty binarne. Metodą dostępu stacji do sieci AS-i jest odpytywanie (ang. polling). Jedną ze stacji jest wyróżniona i pełni rolę zarządcy (ang. master), może to być dodatkowy moduł sterownika PLC. Pozostałe węzły są podległe (ang. slaves).

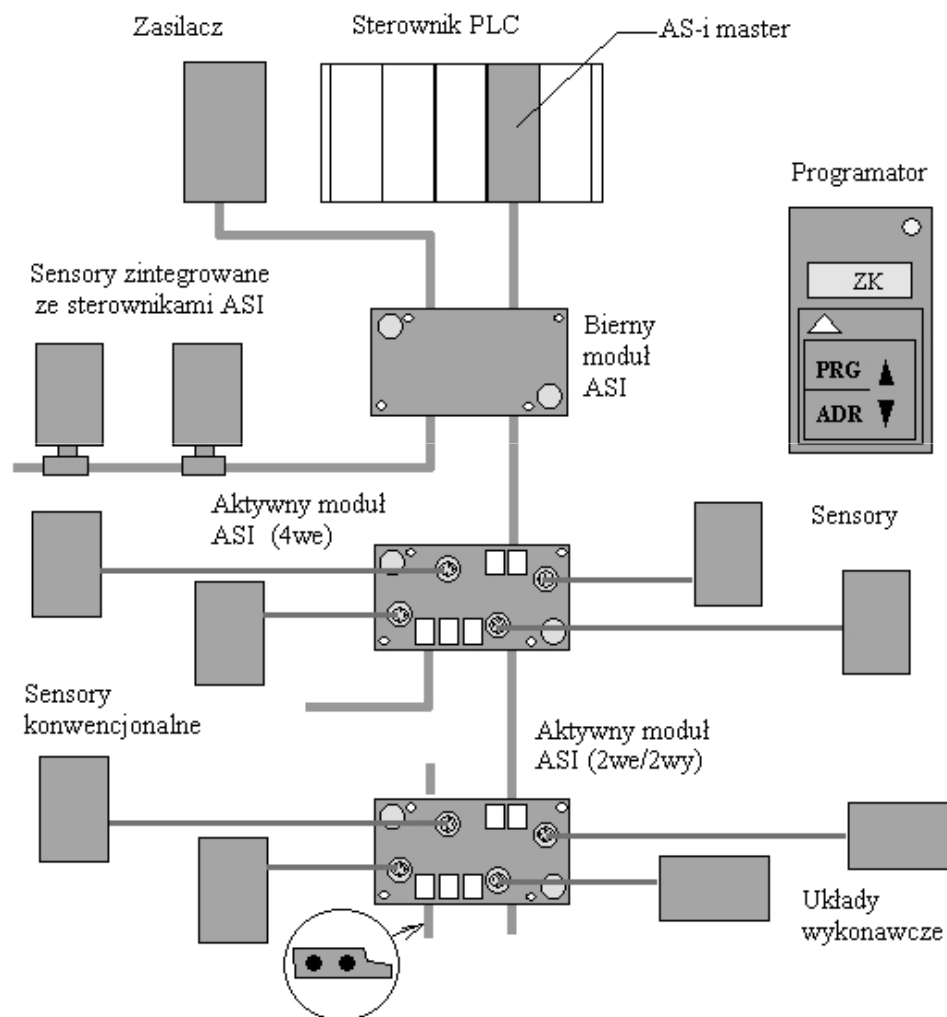
Master posiada pełną listę adresów stacji dołączonych do sieci i odpytuje kolejne slave, przekazując w ten sposób zgodę na transmisję w sieci. Jeden master może obsługiwać do 31 punktów podległych, tzw. modułów, które mogą zawierać układy wyjściowe albo wejściowe.

Węzeł może obsługiwać 4 punkty binarne, tzn. 2 wejścia i 2 wyjścia, 4 wejścia lub 4 wyjścia. Każdy moduł musi mieć swój adres (1-31) zapisany np. w pamięci EEPROM.



Moduły mogą tworzyć sieć o konfiguracji liniowej, gwiazdziej lub drzewiastej. Jako łącze stosowany jest specjalny dwużyłowy ($2 \times 1,5\text{mm}^2$), nieekranowany, profilowany kabel o długości do 100m (300m z regeneratorem). Zalety takiego rozwiązania są następujące: prosty montaż i serwis okablowania, szybki i prawidłowy montaż modułów węzłowych sieci, prosta diagnostyka sieci, łatwa rekonfiguracja oraz duża odporność na uszkodzenia mechaniczne.

Wadą przyjętego kabla jest mniejsza, w porównaniu ze skrętką, odporność na zakłócenia elektromagnetyczne. Zostało to zrekompensowane przez zwiększenie napięcia zasilania (30V DC) oraz zastosowanie dużych prądowych (60mA) sygnałów sterujących.



Budowa sieci ASI

Kabel sieci ASI



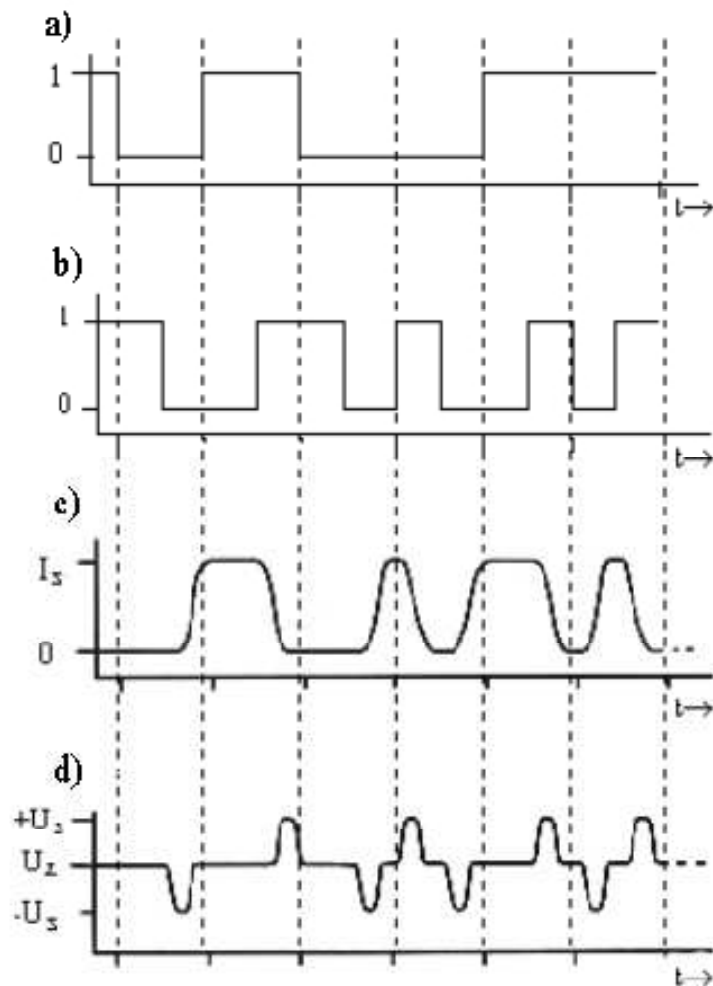


W sieci AS-i realizowana jest transmisja w paśmie podstawowym.

Właściwości transmisyjne naturalnego kodu binarnego nie są najlepsze: ma on stałą składową, liczba kolejnych elementów '0' i '1' nie jest ograniczona i dlatego nie gwarantuje prawidłowej synchronizacji odbiornika. Z drugiej strony, torem transmisyjnym dla AS-i jest para przewodów, którą przesyłane są zarówno dane jak i zasilanie. Powyższe przyczyny powodują konieczność stosowania kodu transmisyjnego. Przyjęto kod PE (ang. Phase Encodage) zwany też kodem Manchester.

Reguła kodowania jest następująca: '1' kodowana jest jako przejście w środku bitu, od poziomu niższego do wyższego, '0' - odwrotnie. Zaletą takiego kodowania jest co najmniej jedno przejście dla każdego bitu.

Przebiegi w nadajniku kontrolera magistrali AS-i pokazano na rysunku.



Modulacja sygnału podczas transmisji

Słowo nadawane, w naturalnym kodzie binarnym (rys.a), kodowane jest według reguł kodu PE (rys.b), a następnie podawane na filtr dolnoprzepustowy. Przebieg ten o ograniczonym paśmie wchodzi na konwerter U/I, na którego wyjściu otrzymujemy prąd I z przedziału (0...60) mA. Takie zmiany prądu są forsowane na magistrali (rys.c). Oddziaływanie między wyjściem nadajnika a specjalnym zasilaczem z odpowiednio dobraną wartością indukcyjności wyjściowej prowadzi do uzyskania naprzemiennej modulacji impulsowej (APM - ang. Alternate Puls Modulation) napięcia zasilającego (rys.d). Modulacja APM odpowiada różniczkowaniu przebiegu z rys.c. Amplituda impulsów napięciowych w torze transmisyjnym wynosi około 2V



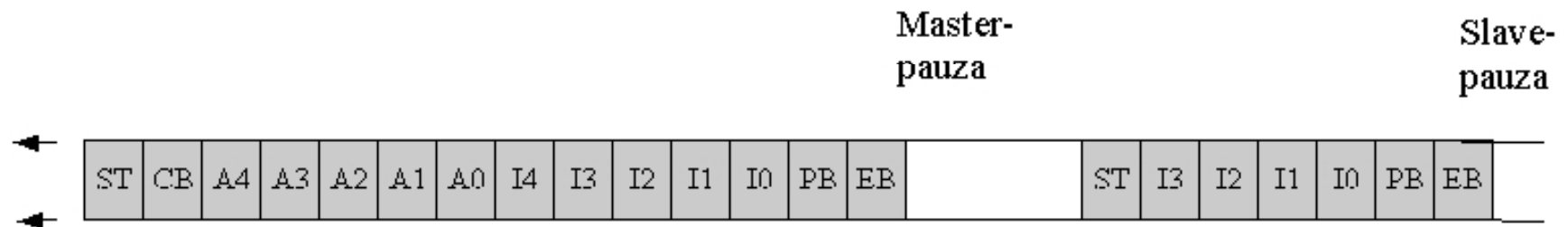
Politechnika Wroclawska

Protokoły transmisji



ASI (ang. Actuator Sensor Interface)

Modulacja sygnału podczas transmisji



- ST - bit startu, zawsze ST=0,
- CB - bit sterujący, 0 - transmisja dane/parametry, 1 - instrukcja sterująca,
- A0...A4 - adres modułu slave, 01H...1FH - AS-i-slave 1 ... AS-i-slave 31,
- I0..I4 - bity informacyjne zależne od typu wywołania,
- PB - bit parzystości, testowane są bity bez ST i EB, 0 - parzysta liczba '1',
- EB - bit stopu, zawsze EB=1.

Format ramki



Master wysyła telegram zaopatrzone w adres modułu slave. Odpowiada wyłącznie wskazany moduł. Ta prosta metoda ściśle determinuje czas dostępu każdego punktu do magistrali.

Dla sieci AS-i przyjęto częstotliwość zegara równą 167kHz co daje czas trwania jednego bitu $6\mu\text{s}$.

Master - pauza wynosi co najmniej 3 i maksymalnie 10 odcinków jednostkowych. Jeśli slave jest zsynchronizowany, wtedy już po trzech odcinkach bitowych przełącza się na nadawanie odpowiedzi. Przy braku synchronizacji wymagane są dwa dodatkowe bity. Jeśli master nie otrzyma bitu startowego odpowiedzi w ciągu 10 bitów pauzy, przechodzi do następnego wywołania.

Slave – pauza trwa tylko jeden odcinek czasowy. Czas cyklu dostępu w sieci AS-i zależy od liczby modułów podrzędnych; przy pełnej obsadzie 31 modułów czas cyklu wynosi około 5ms.



W systemach sterowania istotnym problemem jest zapewnienie wysokiej wiarygodności przekazywanych danych. Zabezpieczenie transmitowanych w sieci AS-i ramek może wydawać się słabe. W rzeczywistości wprowadzenie kodowania transmisyjnego PE i końcowego APM powoduje, że istnieje więcej stref kontroli wynikowego ciągu. W łatwy sposób można wykryć brak transmisji - brak składowej zmiennej

Przyjmuje się następujące kryteria testowania ramek transmitowanych w kodzie APM:

- pierwszy impuls telegramu musi być impulsem ujemnym,
- kolejne pary impulsów muszą mieć przeciwną polaryzację,
- odstęp między sąsiednimi impulsami nie może przekraczać 0,5 okresu zegara,
- w drugiej połowie bitu (w odniesieniu do nadawanego słowa) zawsze musi wystąpić impuls,
- liczba dodatnich impulsów, bez bitów ST i EB musi być parzysta,
- ostatni impuls słowa kodowego musi być dodatni,
- po bicie stopu (EB) nie może być impulsów (dotyczy pauzy).

Następnie po przekodowaniu ciągu z kodu APM na kod PE i dalej na kod binarny, realizowana jest kontrola parzystości odebranej ramki.



Politechnika Wroclawska

Protokoły transmisji



ASI (ang. Actuator Sensor Interface)



Sieć CAN (ang. Controller Area Network) specjalnie została opracowana dla potrzeb pojazdów, w firmie Robert Bosch GmbH,).

Po raz pierwszy na dużą skalę została zastosowana w roku 1990 w samochodach Mercedes S.

W roku 1992 sieć CAN została zatwierdzona przez ISO jako standard międzynarodowy w normach ISO 11898 oraz ISO 11519-2.

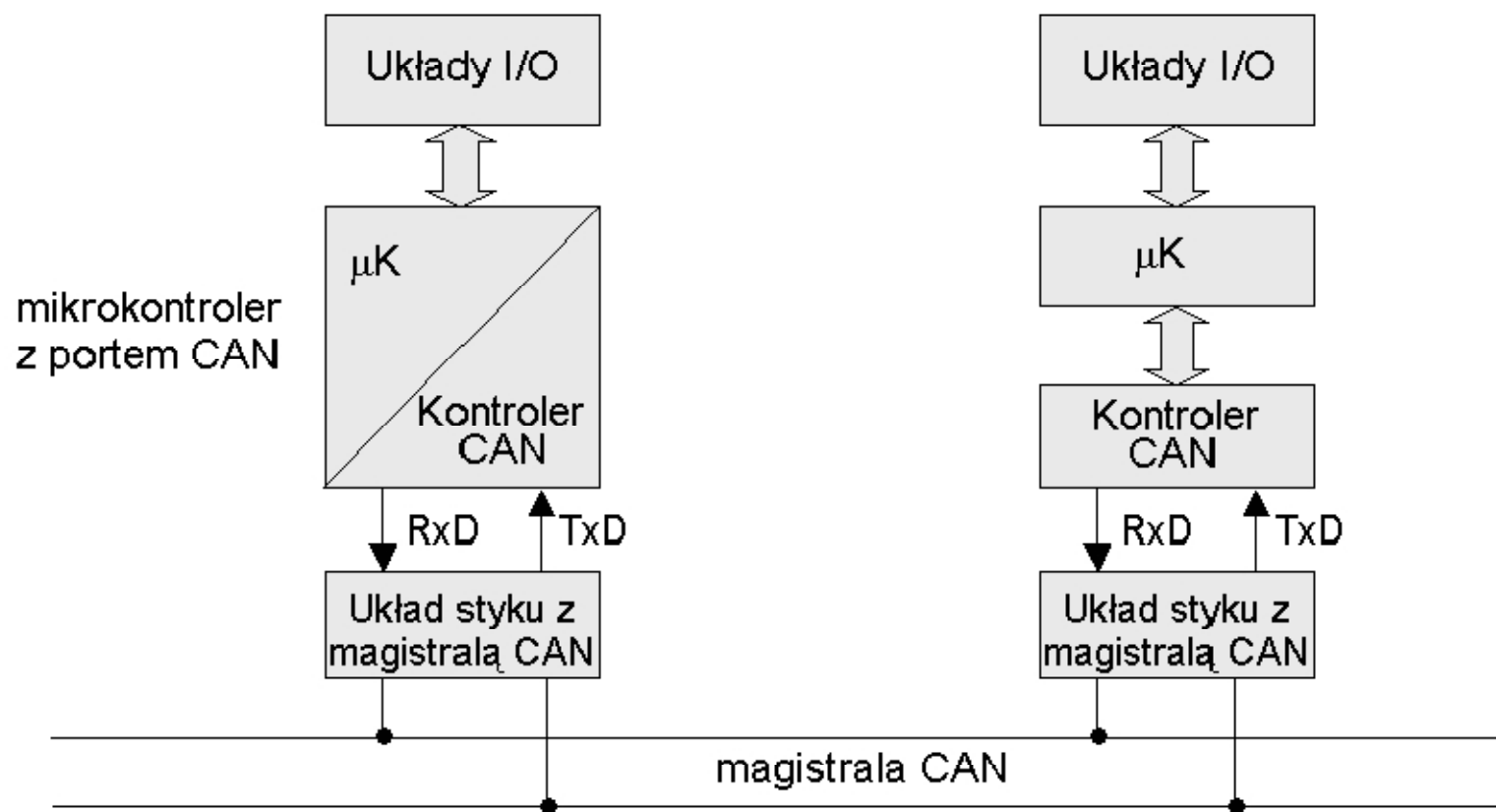
Standard CAN dotyczy głównie warstwy drugiej (DL) oraz częściowo warstwy pierwszej (PL), przy czym pierwszy z dokumentów odnosi się do rozwiązań o dużych prędkościach transmisji a drugi - małych. Ponieważ standard się sprawdził i zyskał bardzo dużą popularność, powstało wiele innych standardów definiujących pozostałe warstwy modelu sieci miejscowej. Przykładowo, standard SAE J1850 przeznaczony jest do zastosowań w samochodach osobowych, a SAE J1939 - dla ciężarówek, autobusów i ciężkich pojazdów specjalnych.



Zestawienie standardów opartych na specyfikacji CAN

Nazwa sieci	Prędkość b/s	Medium transmisyjne	Nad./odb. układ transm.	Obszar zastosowań
SAE J1939-11	250k	2ps, ek	250	Pojazdy ciężarowe i specjalne, autobusy
SAE J1939-12	250k	2ps, ek, zs 12V	250	Rolnictwo - maszyny
SAE J2284	500k	2ps, nek	250	Automatyzacja - duże prędkości,
SAE J2411	25k, 40k	1p	SWC	Automatyzacja - małe prędkości
NMEA2000	62,5k, 125k, 250k, 500k, 1M	2ps, ek, zs	250	Przemysł morski, statki, okręty
Device Net	125k, 250k, 500k	2ps, sk, zs 24V	250	Automatyzacja przemysłu
CANopen	10k, 20k, 50k, 125k, 250k, 500k, 800k, 1M	2ps,ek - opcja, zs - opcja	250	Automatyzacja przemysłu
SDS	125k, 250k, 500k, 1M	2ps, ek, zs - opcja	250	Automatyzacja przemysłu
Fault Tolerant	<125k	2ps, nek	252	Komunikacja ****

Objaśnienia: 2ps - dwuprzewodowa skrętka, 1p - jeden przewód, ek - ekran, nek - bez ekranu, zs - przewody zasilania



Organizacja węzłów sieci CAN



Kontrolery możemy podzielić na dwie podstawowe grupy: specyficzne - realizujące wyłącznie zadania CAN oraz uniwersalne, oparte na znanych mikrokontrolerach wyposażonych w kanały CAN. W pierwszej grupie występuje kilkanaście układów. Do najpopularniejszych należą SJA1000 (Philips), AN82257 (Intel) a także SAE81C91 (Siemens).

Druga grupa układów jest znacznie bardziej liczna. Praktycznie każdy z producentów mikroprocesorów ma w swojej ofercie co najmniej jeden uniwersalny mikrokontroler z kanałami CAN. Przykłady: PIC16C185 (Microchip), 8X196CA (Intel), M37632MF (Mitsubishi), SABC164 (Siemens) czy P87C592 (Philips).

Na rysunku przedstawiono dwa ogólne rozwiązania węzłów sieci CAN. Jedno wykorzystuje uniwersalny mikrokontroler z dostępnymi portami CAN. Jest to rozwiązanie szczególnie korzystne dla dużej serii węzłów.

W konstrukcjach jednostkowych, prototypowych lepsza jest druga konfiguracja oparta na specjalizowanych kontrolerach CAN i dowolnych, uniwersalnych mikrokontrolerach realizujących zadania warstwy aplikacyjnej.

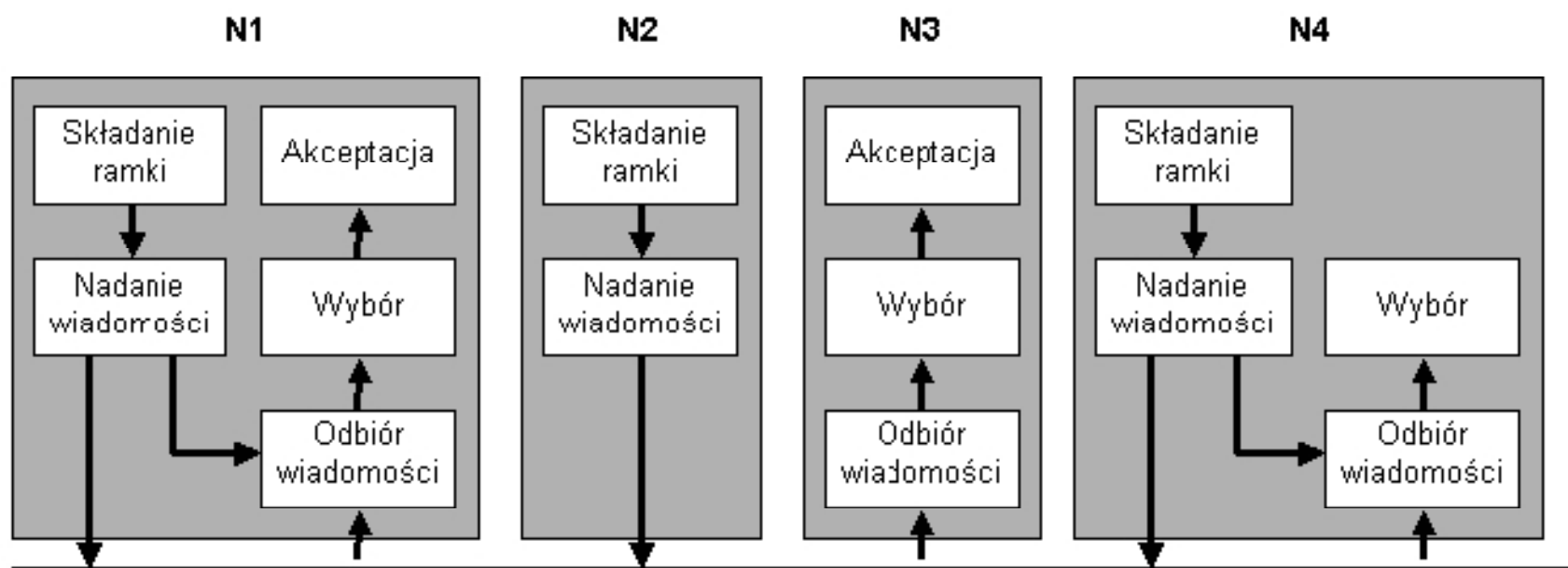


Magistrala z obu stron powinna być zakończona dopasowującymi impedancjami (terminatory - ok. 120Ω). Jedną z linii oznaczana jest **CAN-H** a druga **CAN-L**. Wyjścia układów sterujących magistralą mogą znajdować się w trzech stanach logicznych: stan wysokiej impedancji, **stan dominujący** i **stan recesywny**. Stan dominujący oznacza, że różnica napięć CAN-H - CAN-L wynosi nie mniej niż 0,9V a dla stanu recesywnego napięcie różnicowe nie przekracza 0,5V.

W przypadku jednoczesnego podania bitu dominującego i bitu recesywnego przez nadajniki dwóch różnych węzłów, na magistrali ustali się poziom dominujący.

Opracowanych zostało wiele monolitycznych układów nadawczo-odbiorczych realizujących styk z fizyczną magistralą jak:

CF150B (Bosch), MTC3054 (Alcatel Mietec), PCA82C250, PCA82C251, PCA82C252, TJA1053 (Philips), TLE6252G (Siemens), Si9200EY, B10011S (Temic Siliconix), SN75LBC031, SN65LBC031 (Texas), UC5350 (Unitrode) itd.



Fragment przykładowej sieci CAN



Zgodnie ze specyfikacją Bosch'a warstwa łącza danych dzieli się na dwie podwarstwy: obiektową i transferową. Do zadań podwarstwy obiektowej należy filtracja wiadomości oraz obsługa wiadomości i statusu. Podwarstwa transferu zajmuje się przygotowaniem ramki wiadomości, zatwierdzaniem i rozpakowaniem wiadomości, detekcją i sygnalizacją błędów, arbitrażem oraz prędkością transmisji i zależnościami czasowymi.

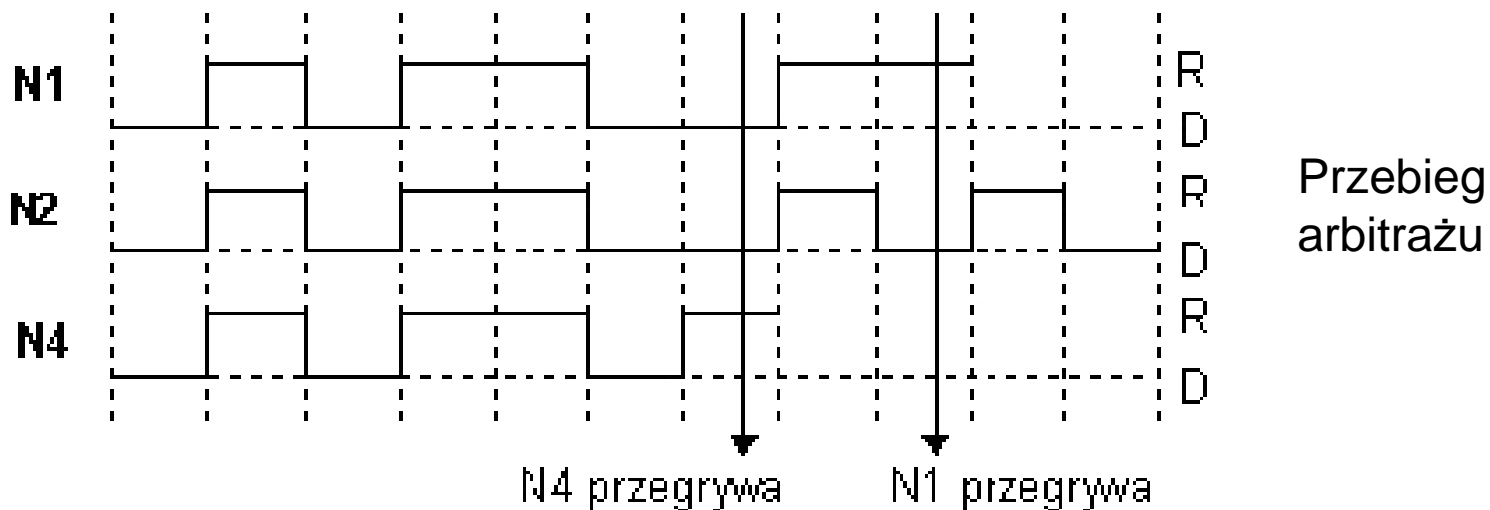
CAN jest protokołem typu **M-M** (ang. multi-master) to znaczy, że poszczególne węzły sieci mogą jednocześnie żądać dostępu do magistrali. Taka sytuacja wymaga arbitrażu. Przyjęto modyfikację metody CSMA/CD (ang. Carrier Sense Multiple Access with Collision Detect), polegającą na tzw. **nieniszczącym arbitrażu**, który oznacza, że w przypadku wystąpienia kolizji dostęp do magistrali uzyskuje wiadomość o najwyższym priorytecie.

Dokładniej zasadę arbitrażu wyjaśni przykład. Uwzględnione zostaną cztery węzły sieci o organizacji jak na rysunku.



Identyfikatory wiadomości nadawanych i odbieranych w przykładowej sieci CAN

N1 (nad.+ odb.) - ident.wiad.nadawanych	$01011001100_B = 716_D = 2CC_H$,
ident.wiad.odbieryanych	$01011001010_R = 714_D = 2CA_H$,
N2 (tylko nad.) - ident.wiad.nadawanych	$01011001010_B = 714_D = 2CA_H$,
N3 (tylko odb.) - ident.wiad.odbieryanych	$01011001xxx_B = 712_D..719_D = 2C8_H..2CF_H$,
N4 (nad.+ odb.) - ident.wiad.nadawanych	$01011010010_B = 722_D = 2D2_H$,
ident.wiad.odbieryanych	$01011011xx1_B = 729_D..735_D = 2D9_H..2DF_H$.



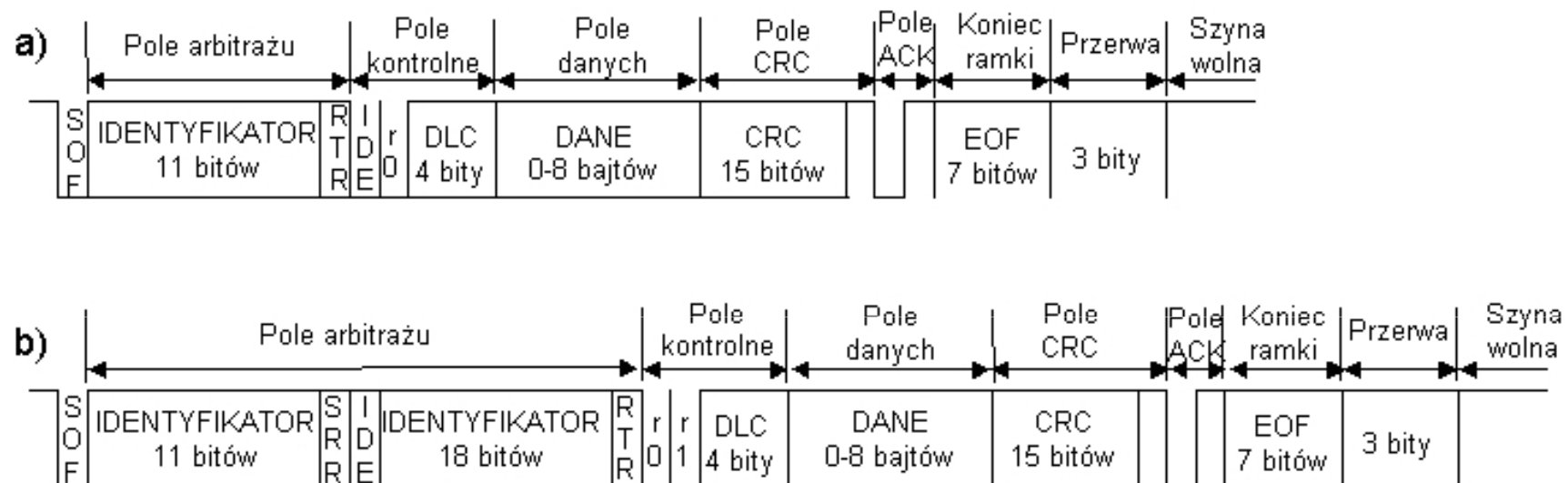


W sieci CAN podstawą transmisji jest wymiana wiadomości. Dany węzeł może odbierać i nadawać wiele wiadomości. Odbiorniki N3 i N4 akceptują po kilka wiadomości. Załóżmy, że jednocześnie startują węzły N1, N2 i N4. Przebieg arbitrażu pokazano na rysunku.

W wersji podstawowej CAN, identyfikator wiadomości zawiera 11 bitów (pole arbitrażu). Pierwsza różnica pojawia się dla N4, który wysyła bit recesywny a N1 i N2 generują bity dominujące. Na magistrali ustala się stan dominujący. N4 przegrywa - następuje wyłączenie nadajnika. Nadal rywalizują N1 i N2.

Dostęp do magistrali uzyskuje wiadomość nadawana przez węzeł N2.

Widać następującą zależność: **im niższy numer identyfikatora tym wyższy priorytet wiadomości**. Węzły N1, N2, N4 nadają wiadomości, które są odbierane przez N1 i N3. Węzeł N4 analizuje również wiadomości na magistrali ale ich nie akceptuje - oczekuje na inne wiadomości (o identyfikatorach 729D..735D)



Formaty ramek wiadomości: a) ramka formatu standardowego (CAN 2.0A),
b) ramka formatu rozszerzonego CAN 2.0B),



Wiadomości przekazywane są przy pomocy ramek w formacie standardowym (CAN 2.0A) lub w formacie rozszerzonym (CAN 2.0B). Różnią się identyfikatorem - 11 lub 29 bitów, co pozwala na rozróżnienie bardzo dużej liczby wiadomości. Ograniczenie wynika jedynie z warstwy fizycznej.

W praktyce znaczna długość identyfikatorów pozwala na kodowanie dodatkowych informacji dotyczących wiadomości lub węzłów (typ wiadomości, rodzaj sensora czy np. kod producenta).

W sieci CAN nie ma adresowania konkretnych węzłów, a każda wysłana wiadomość może być odebrana przez dowolny moduł. Akceptacja danych zależy wyłącznie od ustawienia układu wyboru w odbiorniku węzła.

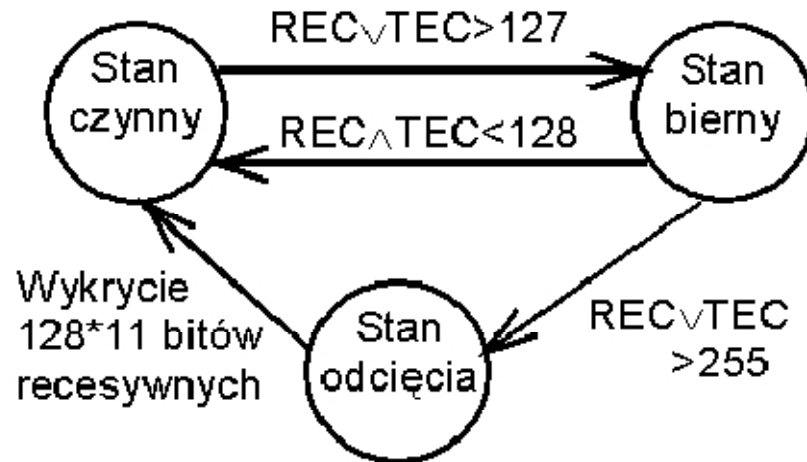


Ramka rozpoczyna się od bitu dominującego SOF (ang. start of frame). Dalej następuje pole arbitrażu, które jest wykorzystywane przy rozstrzygnięciu ewentualnych konfliktów z dostępem do sieci oraz służy do identyfikacji wiadomości w sieci.

Bit RTR rozróżnia ramkę z danymi (bit dominujący) od ramki zdalnej (bit recesywny). Format ramki określany jest przez bit IDE (bit dominujący - ramka standardowa, bit recesywny - ramka rozszerzona).

Bit SRR wysyłany jest jako recesywny. Bity rezerwowe r0 i r1 są ustawione jako dominujące. Pole kontrolne definiuje liczbę bajtów danych. Zawartość DCL w zakresie 0..7 bezpośrednio równa jest liczbie bajtów, natomiast wartość 8..15 oznacza 8 bajtów. To umożliwia kodowanie dodatkowych informacji. Szczególne znaczenie ze względu na wiarygodność danych ma pole CRC. Bit 16-ty kończący to pole (ogranicznik) jest recesywny. Kontrola dla pól od SOF do pola danych (jeśli występuje) włącznie, realizowana jest przy pomocy kodu cyklicznego. Pole potwierdzenia (ACK) zawiera 2 bity: ACK i bit ogranicznika. Po wysłaniu ramki nadawca wysyła 2 recesywne bity pola ACK. Odbiornik, który poprawnie odebrał wiadomość odpowiada w tym samym czasie dominującym bitem ACK. Dzięki temu nadawca uzyskuje potwierdzenie, że przynajmniej jeden węzeł poprawnie odebrał wiadomość.

Pole EOF składa się z 7 recesywnych bitów kończących ramkę.



Typy wykrywanych błędów:

- stanu bitu,
- szpicowania bitami,
- kontroli kodowej (CRC),
- potwierdzenia (ACK),
- formatu ramki

Stany węzłów przy obsłudze błędów CAN



Ważnym zadaniem warstwy łącza danych jest obsługa błędów. Gdy którykolwiek z węzłów wykryje błąd generowana jest specjalna ramka błędu, zawierająca flagę błędu (6 bitów dominujących). Bity dominujące flagi błędu powodują nadpisanie uszkodzonej ramki i wymuszają powtórzenie transmisji. Każdy z węzłów przeprowadza kontrolę bitów.

Błąd stanu bitu występuje, jeśli nadajnik wyśle bit dominujący, a odbierze recesywny albo wyśle recesywny, a odbierze dominujący. Wyjątkiem od tej zasady jest sekwencja arbitrażu.

Błąd szpikowania zachodzi wtedy, gdy w ciągu bitów między SOF a ogranicznikiem CRC pojawi się więcej niż 5 takich samych bitów - odbiornik stwierdza naruszenie zasady szpikowania.

Odbiornik na podstawie ciągu CRC sprawdza czy w ramce nie nastąpiło przekłamanie.

Mimo znacznej odległości Hamminga równej 6, kod wykorzystywany jest wyłącznie do wykrywania błędów a nie do korekcji.

Odbiornik, który poprawnie odbierze dane ustawia bit ACK jako dominujący. W ten sposób nadajnik uzyskuje potwierdzenie poprawnej transmisji; w przeciwnym wypadku nadajnik wysyła ramkę błędu.

W ramce występują bity, które mają znane wartości. Zmiana tych bitów oznacza błąd formatu ramki.



W celu odróżnienia błędów chwilowych od trwałych kontrolery CAN wyposażone są w dwa liczniki: licznik błędów odbioru (**REC** - Receive Error Counter) oraz licznik błędów nadawania (**TEC** – Transmit Error Counter).

Liczniki zwiększają swoją zawartość przy błędzie odpowiedniej operacji i zmniejszają ją przy poprawnej operacji. W zależności od stanu licznika może zmieniać się stan węzła. Węzły mogą znajdować się w trzech stanach – stan czynny, stan bierny, stan odcięcia.

Stan czynny ($REC=TEC<128$) jest normalnym stanem węzłów - są w pełni aktywne. Gdy sieć działa bez zakłóceń $REC=TEC=0$. W stanie pasywnym ($127<REC<TEC<255$) węzły nadal nadają i odbierają wiadomości. Różnica polega na tym, że w przypadku błędu, węzeł czynny wysyła czynną ramkę błędu (z czynną flagą błędu - 6 bitów dominujących) a węzeł bierny wysyła bierną ramkę błędu (z bierną flagą błędu - 6 bitów recesywnych).

Przekroczenie zawartości $REC<TEC>255$ powoduje przejście węzła do stanu odcięcia.