

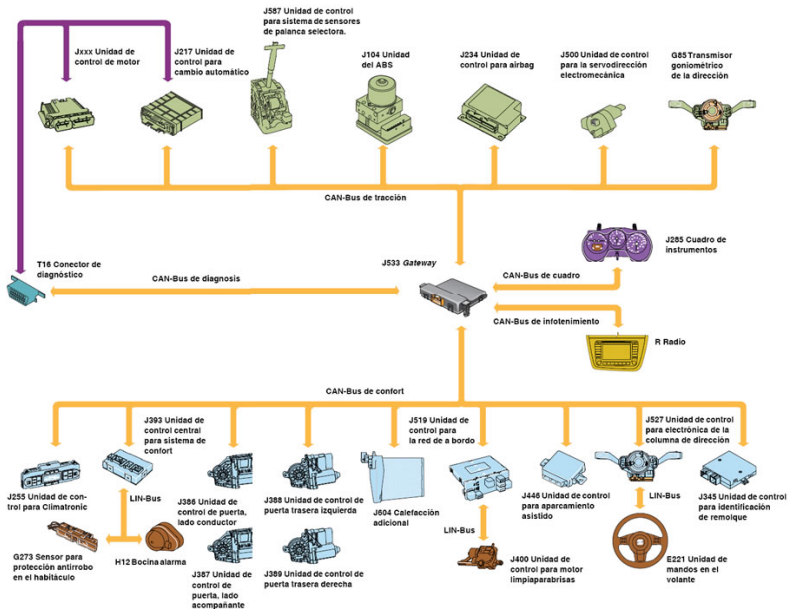
# Systemy wbudowane - wykład kolejny

Przemek Błaśkiewicz

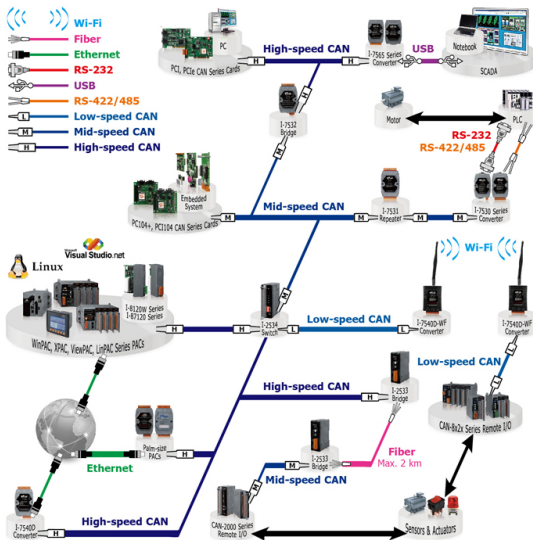
20 marca 2019



# CAN – controller area network

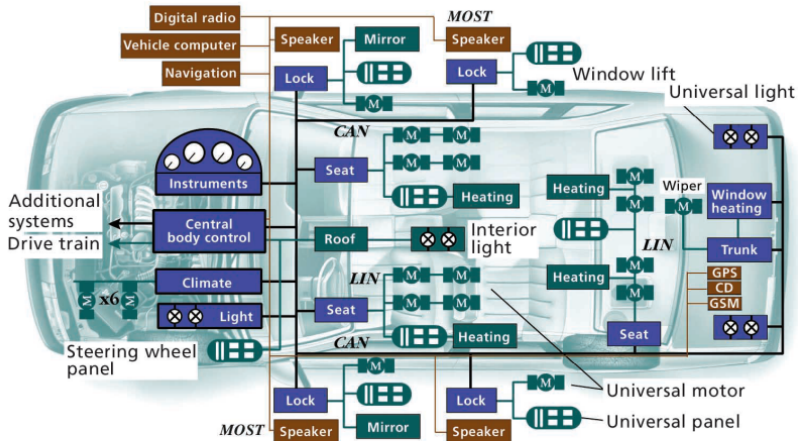


# CAN – controller area network



[http://www.icpdas.com/root/product/solutions/industrial\\_communication/fieldbus/can\\_bus/can\\_intro.html](http://www.icpdas.com/root/product/solutions/industrial_communication/fieldbus/can_bus/can_intro.html)

# CAN – controller area network



- CAN Controller area network
- GPS Global Positioning System
- GSM Global System for Mobile Communications
- LIN Local interconnect network
- MOST Media-oriented systems transport

Link do artykułu na stronie. Polecam! P. Błażkiewicz



- transmisja typu broadcast (semi dupleks) - multi-master

- transmisja typu broadcast (semi duplex) - multi-master
- transmisja asynchroniczna, szeregowo, oparta na ramkach



- transmisja typu broadcast (semi duplex) - multi-master
- transmisja asynchroniczna, szeregowo, oparta na ramkach
- CSMA/CA - carrier sense multiple acces / collision avoidance

- transmisja typu broadcast (semi duplex) - multi-master
- transmisja asynchroniczna, szeregowo, oparta na ramkach
- CSMA/CA - carrier sense multiple acces / collision avoidance
- detekcja/mitygacja błędów transmisji/urządzeń

- transmisja typu broadcast (semi duplex) - multi-master
- transmisja asynchroniczna, szeregowo, oparta na ramkach
- CSMA/CA - carrier sense multiple acces / collision avoidance
- detekcja/mitygacja błędów transmisji/urządzeń
- przepustowość od 10kbps (do 5 km) do 1Mbps na odległość do  $\approx 25$  m (ok. 50% zawartości ramki to dane)

- transmisja typu broadcast (semi duplex) - multi-master
- transmisja asynchroniczna, szeregowo, oparta na ramkach
- CSMA/CA - carrier sense multiple acces / collision avoidance
- detekcja/mitygacja błędów transmisji/urządzeń
- przepustowość od 10kbps (do 5 km) do 1Mbps na odległość do  $\approx 25$  m (ok. 50% zawartości ramki to dane)
- standard CAN definiuje *warstwę fizyczną i łącza danych*

- transmisja typu broadcast (semi duplex) - multi-master
- transmisja asynchroniczna, szeregowo, oparta na ramkach
- CSMA/CA - carrier sense multiple acces / collision avoidance
- detekcja/mitygacja błędów transmisji/urządzeń
- przepustowość od 10kbps (do 5 km) do 1Mbps na odległość do  $\approx 25$  m (ok. 50% zawartości ramki to dane)
- standard CAN definiuje *warstwę fizyczną i łącza danych*
  - co to jest "bit", jakie medium transmisyjne)

- transmisja typu broadcast (semi duplex) - multi-master
- transmisja asynchroniczna, szeregowo, oparta na ramkach
- CSMA/CA - carrier sense multiple acces / collision avoidance
- detekcja/mitygacja błędów transmisji/urządzeń
- przepustowość od 10kbps (do 5 km) do 1Mbps na odległość do  $\approx 25$  m (ok. 50% zawartości ramki to dane)
- standard CAN definiuje *warstwę fizyczną i łącza danych*
  - co to jest "bit", jakie medium transmisyjne)
  - kontrola dostępu do medium (MAC) i obsługa błędów (*error detection/confinement*, kształt wiadomości/ramki)

- transmisja typu broadcast (semi duplex) - multi-master
- transmisja asynchroniczna, szeregowo, oparta na ramkach
- CSMA/CA - carrier sense multiple acces / collision avoidance
- detekcja/mitygacja błędów transmisji/urządzeń
- przepustowość od 10kbps (do 5 km) do 1Mbps na odległość do  $\approx 25$  m (ok. 50% zawartości ramki to dane)
- standard CAN definiuje *warstwę fizyczną i łącza danych*
  - co to jest "bit", jakie medium transmisyjne
  - kontrola dostępu do medium (MAC) i obsługa błędów (*error detection/confinement*, kształt wiadomości/ramki)
  - na tym zbudowane warstwy obiektów (rodzaje wiadomości, filtrowanie, routing) i aplikacji (np. CANOpen, DeviceNet, etc.)

## Warstwa fizyczna

Definiuje poziomy sygnałów (napięcia) oraz jaki jest stan dominujący. Określa sposób kodowania poszczególnych bitów oraz określa synchronizację magistrali.



## Warstwa fizyczna

Definiuje poziomy sygnałów (napięcia) oraz jaki jest stan dominujący. Określa sposób kodowania poszczególnych bitów oraz określa synchronizację magistrali.

## Warstwa łącza danych

Definiuje strukturę ramki, sposoby sygnalizacji błędów i mechanizm MAC (dostępu do medium).

- wymagany jest MAC na poziomie bitowym

- wymagany jest MAC na poziomie bitowym
- bity dominujące i recesywne

- wymagany jest MAC na poziomie bitowym
- bity dominujące i recesywne
- mechaniczne aspekty (wtyki, medium) nie określone w standardzie

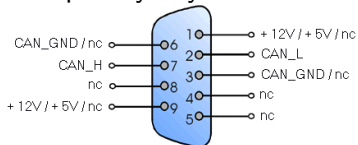
- wymagany jest MAC na poziomie bitowym
- bity dominujące i recesywne
- mechaniczne aspekty (wtyki, medium) nie określone w standardzie

często:

- wymagany jest MAC na poziomie bitowym
- bity dominujące i recesywne
- mechaniczne aspekty (wtyki, medium) nie określone w standardzie

często:

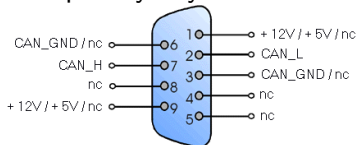
- 9-cio pinowy wtyk D-sub: CAN-Lo, GND, CAN-Hi, CAN-V+



- wymagany jest MAC na poziomie bitowym
- bity dominujące i recesywne
- mechaniczne aspekty (wtyki, medium) nie określone w standardzie

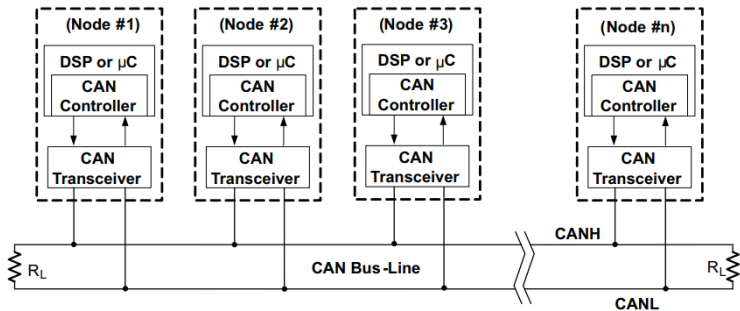
często:

- 9-cio pinowy wtyk D-sub: CAN-Lo, GND, CAN-Hi, CAN-V+



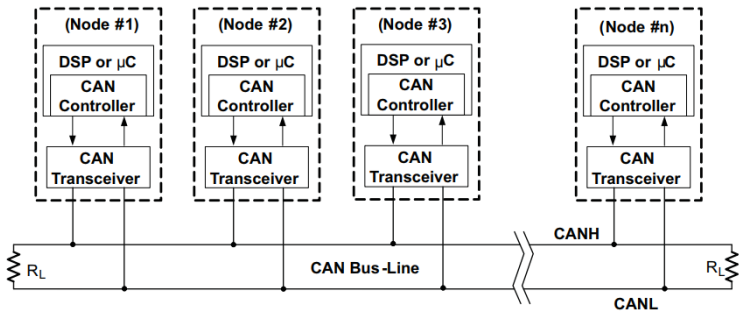
- wspólna szyna zasilająca +5V (3.3V opcjonalnie) oraz GND

# CAN - magistrala





# CAN - magistrala

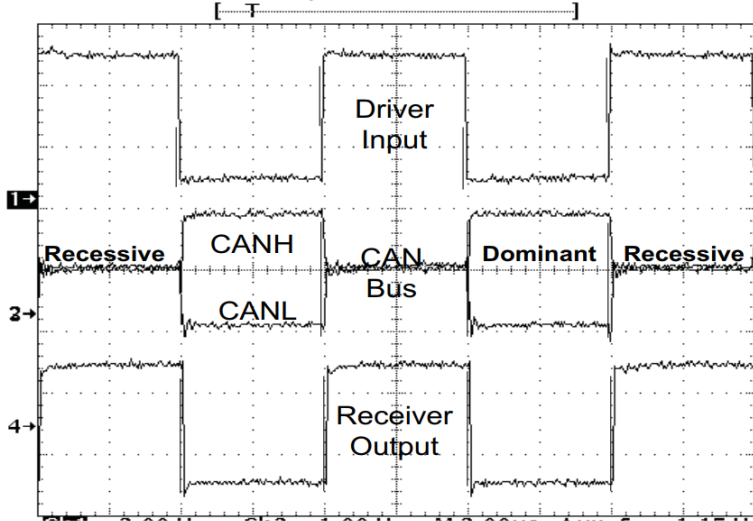


## Sygnalizacja różnicowa

Sygnał "bez niczego" jest utrzymywany w okolicach 2.5V. CAN-Hi jest podciągany o 1V w górę, a CAN-Lo o 1V w dół, co daje 2V różnicy między parą linii sygnałowych.

# CAN - magistrala

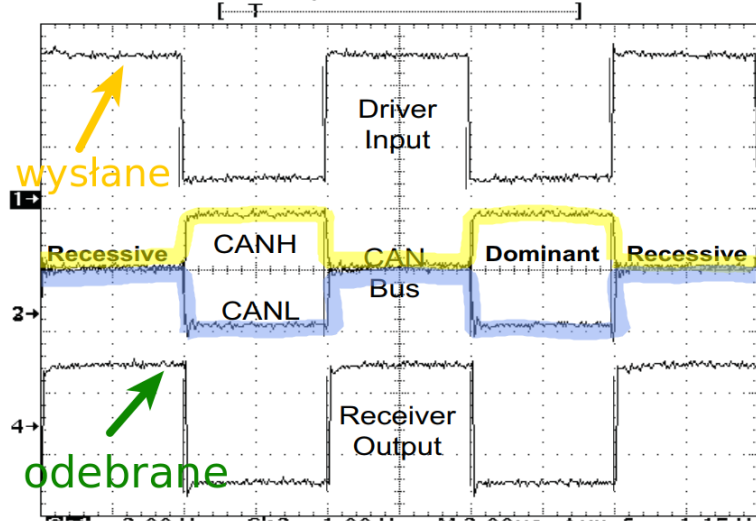
TEK Run: 25.0MS/s Sample



Steve Corrigan: *Introduction to the Controller Area Network (CAN)*, SLOA101A, [www.ti.com](http://www.ti.com)

# CAN - magistrala

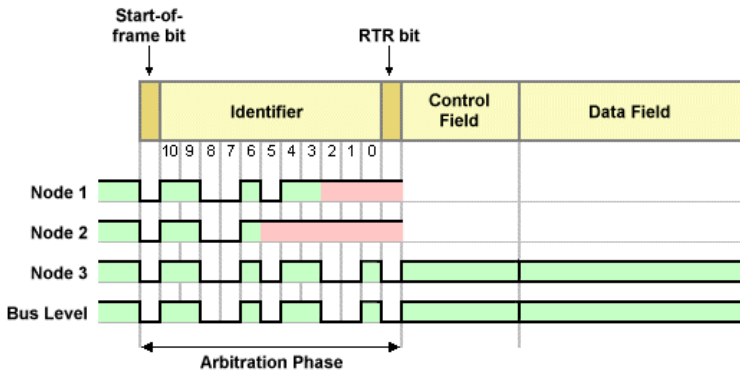
TEK Run: 25.0MS/s Sample



Steve Corrigan: *Introduction to the Controller Area Network (CAN)*, SLOA101A, [www.ti.com](http://www.ti.com)



# CAN - arbitraz magistrali



**Poziom wiadomości:**

## Poziom wiadomości:

- CRC (15 bitów + 1 bit odstępu)

## Poziom wiadomości:

- CRC (15 bitów + 1 bit odstępu)
- ACK (1 bit potwierdzenia + 1 bit odstępu)



## Poziom wiadomości:

- CRC (15 bitów + 1 bit odstępu)
- ACK (1 bit potwierdzenia + 1 bit odstępu)
- bity recesywne:  
CRC-delim, ACK-delim, SOF, EOF

## Poziom wiadomości:

- CRC (15 bitów + 1 bit odstępu)
- ACK (1 bit potwierdzenia + 1 bit odstępu)
- bity recesywne:  
CRC-delim, ACK-delim, SOF, EOF

## Poziom bitów:

## Poziom wiadomości:

- CRC (15 bitów + 1 bit odstępu)
- ACK (1 bit potwierdzenia + 1 bit odstępu)
- bity recesywne:  
CRC-delim, ACK-delim, SOF, EOF

## Poziom bitów:

- monitorowanie magistrali danych (brak ACK, adres)

## Poziom wiadomości:

- CRC (15 bitów + 1 bit odstępu)
- ACK (1 bit potwierdzenia + 1 bit odstępu)
- bity recesywne:  
CRC-delim, ACK-delim, SOF, EOF

## Poziom bitów:

- monitorowanie magistrali danych (brak ACK, adres)
- bit stuffing

Typy ramek:

- danych – faktycznie przesyła dane

Typy ramek:

- danych – faktycznie przesyła dane
- zdalnego wywołania – żądanie przesłania danych

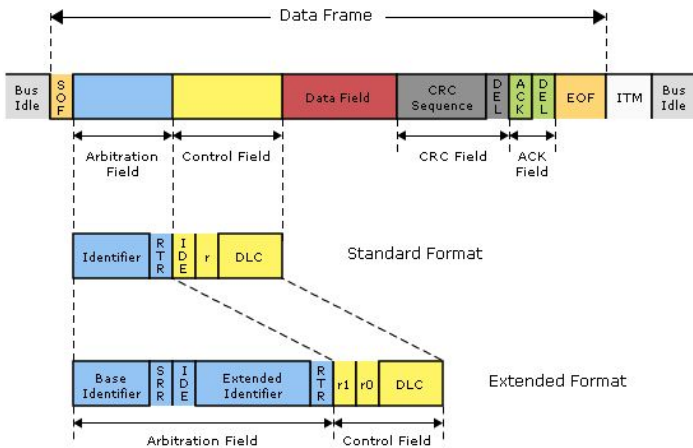
Typy ramek:

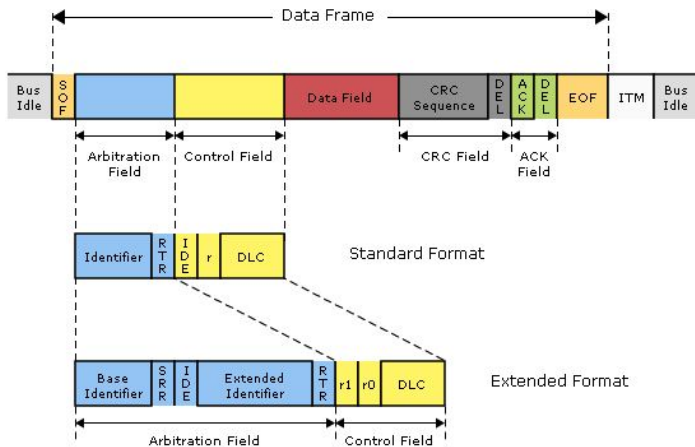
- danych – faktycznie przesyła dane
- zdalnego wywołania – żądanie przesłania danych
- błędów – w przypadku wystąpienia błędu

## Typy ramek:

- danych – faktycznie przesyła dane
- zdalnego wywołania – żądanie przesłania danych
- błędów – w przypadku wystąpienia błędu
- przepełnienia – do uzyskania opóźnienia między ramką danych lub zdalnego wywołania

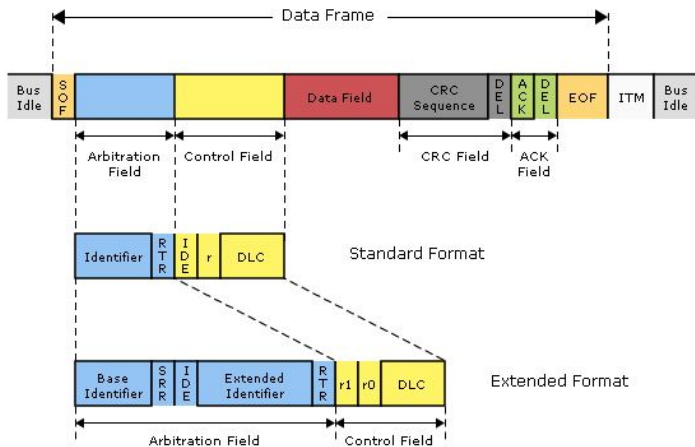






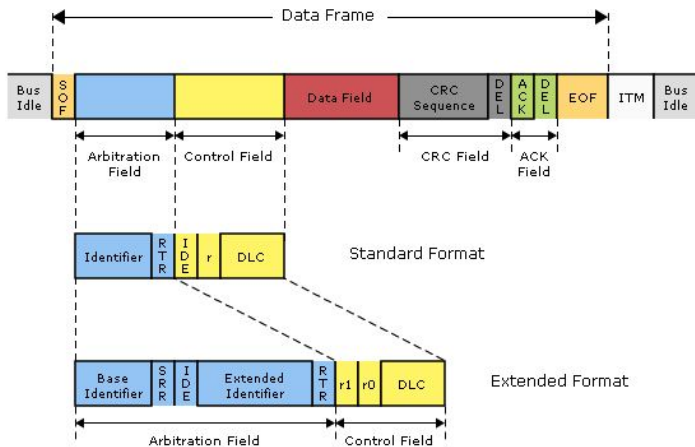
## Format standard

RTR	<i>remote transmission request</i> – 0 dla ramek danych, 1 dla zdalnego wywołania
IDE	<i>id extension</i> – 0 dla ramek formatu podstawowego, 1 dla rozszerzonego
DLC	<i>data length</i> – długość danych (0-8 bajtów)
CRC-del	recesywny bit (2)
ACK-slot	odbiorca ma szansę wysłać 0 (dominujący), bo nadawca wysłał 1 (recesywny)
ACK-del	recesywny bit (1)
EOF	<i>end of frame</i> - recesywny bit (1)



## Format extended

SRR | *substitute remote request – recesywny bit (1)*  
 IDE | *identifier extension bit – recesywny bit (1)*



## Ramka zdalna

$RTR = 1 \rightarrow$  przegrywa arbitraż kiedy jednocześnie jest ramka danych z tym samym identyfikatorem

## Podstawowa ramka błędu

Składa się z 6 bitów dominujących (ERROR FLAG) oraz 8 bitów recesywnych (ERROR DELIMITER). Bity dominujące (dla ERROR FLAG) wysyła stacja będąca aktywną (tj. dla której nie określono błędnego działania). Bity recesywne wysyła stacja w stanie *passive error*.

## Podstawowa ramka błędu

Składa się z 6 bitów dominujących (ERROR FLAG) oraz 8 bitów recesywnych (ERROR DELIMITER). Bity dominujące (dla ERROR FLAG) wysyła stacja będąca aktywną (tj. dla której nie określono błędnego działania). Bity recesywne wysyła stacja w stanie *passive error*.

Jest to świadome naruszenie zasady bit-stuffing (poniżej). Każdy węzeł, który rozpoznaje taki błąd, wysyła na magistralę swoją ramkę błędu.

## Podstawowa ramka błędu

Składa się z 6 bitów dominujących (ERROR FLAG) oraz 8 bitów recesywnych (ERROR DELIMITER). Bity dominujące (dla ERROR FLAG) wysyła stacja będąca aktywną (tj. dla której nie określono błędnego działania). Bity recesywne wysyła stacja w stanie *passive error*.

Jest to świadome naruszenie zasady bit-stuffing (poniżej). Każdy węzeł, który rozpoznaje taki błąd, wysyła na magistralę swoją ramkę błędu.

Dlatego w sumie bitów dominujących (ERROR FLAG) może być 6-12, plus 8 bitów recesywnych.

## Podstawowa ramka błędu

Składa się z 6 bitów dominujących (ERROR FLAG) oraz 8 bitów recesywnych (ERROR DELIMITER). Bity dominujące (dla ERROR FLAG) wysyła stacja będąca aktywną (tj. dla której nie określono błędnego działania). Bity recesywne wysyła stacja w stanie *passive error*.

Jest to świadome naruszenie zasady bit-stuffing (poniżej). Każdy węzeł, który rozpoznaje taki błąd, wysyła na magistralę swoją ramkę błędu.

Dlatego w sumie bitów dominujących (ERROR FLAG) może być 6-12, plus 8 bitów recesywnych.

Po czym następuje przerwa międzyramkowa (3 bity).



## Podstawowa ramka błędu

Składa się z 6 bitów dominujących (ERROR FLAG) oraz 8 bitów recesywnych (ERROR DELIMITER). Bity dominujące (dla ERROR FLAG) wysyła stacja będąca aktywną (tj. dla której nie określono błędnego działania). Bity recesywne wysyła stacja w stanie *passive error*.

Jest to świadome naruszenie zasady bit-stuffing (poniżej). Każdy węzeł, który rozpoznaje taki błąd, wysyła na magistralę swoją ramkę błędu.

Dlatego w sumie bitów dominujących (ERROR FLAG) może być 6-12, plus 8 bitów recesywnych.

Po czym następuje przerwa międzyramkowa (3 bity).

Ile to może trwać? (przykład na tablicy)



Każde urządzenie ma liczniki TxE, RxE (błędy przy nadawaniu, odbiorze).

Każde urządzenie ma liczniki TxE, RxE (błędy przy nadawaniu, odbiorze).

- ❶ błąd przy nadawaniu  $\rightarrow$  TxE += 8 (TxE- - jeśli OK)\*

Każde urządzenie ma liczniki TxE, RxE (błędy przy nadawaniu, odbiorze).

- ❶ błąd przy nadawaniu  $\rightarrow$  TxE += 8 (TxE- - jeśli OK)\*
- ❷ błąd przy odbiorze  $\rightarrow$  RxE += 1 (RxE- - jeśli OK)\*

Każde urządzenie ma liczniki TxE, RxE (błędy przy nadawaniu, odbiorze).

- ❶ błąd przy nadawaniu  $\rightarrow$  TxE += 8 (TxE- - jeśli OK)\*
- ❷ błąd przy odbiorze  $\rightarrow$  RxE += 1 (RxE- - jeśli OK)\*
- ❸ TxE > 127 lub RxE > 127  $\rightarrow$  *passive error state*

Każde urządzenie ma liczniki TxE, RxE (błędy przy nadawaniu, odbiorze).

- ❶ błąd przy nadawaniu  $\rightarrow$  TxE += 8 (TxE- - jeśli OK)\*
- ❷ błąd przy odbiorze  $\rightarrow$  RxE += 1 (RxE- - jeśli OK)\*
- ❸ TxE > 127 lub RxE > 127  $\rightarrow$  *passive error state*
- ❹ TxE > 255 lub RxE > 255  $\rightarrow$  *bus-off state*

Każde urządzenie ma liczniki TxE, RxE (błędy przy nadawaniu, odbiorze).

- ❶ błąd przy nadawaniu  $\rightarrow$  TxE += 8 (TxE- - jeśli OK)\*
- ❷ błąd przy odbiorze  $\rightarrow$  RxE += 1 (RxE- - jeśli OK)\*
- ❸ TxE > 127 lub RxE > 127  $\rightarrow$  *passive error state*
- ❹ TxE > 255 lub RxE > 255  $\rightarrow$  *bus-off state*
- ❺ RESET oraz poprawnie przesłano 128 x 11 bitów  $\rightarrow$  *active error state*



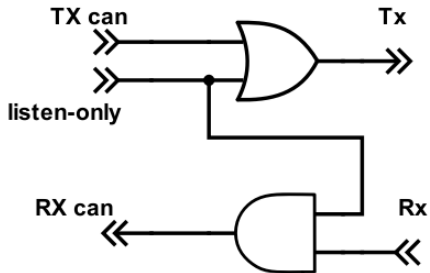
- ① domyślny (default) bit-rate

- ① domyślny (default) bit-rate
- ② automatyczna detekcja

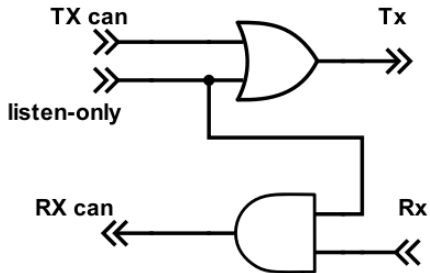
- ① domyślny (default) bit-rate
- ② automatyczna detekcja
  - pomiar czasu trwania pojedynczego bitu (jest taki?)

- ① domyślny (default) bit-rate
- ② automatyczna detekcja
  - pomiar czasu trwania pojedynczego bitu (jest taki?)
  - cichutkie podsłuchiwanie (niektórzy nie potrafią...)

# Listen-only dla nieumiejących



# Listen-only dla nieumiejących



A jak już to umiemy to...

Po każdym 5 takich samych bitach automatycznie wtawiany jest bit przeciwny:

Po każdym 5 takich samych bitach automatycznie wtawiany jest bit przeciwny:

- by utrzymać synchronizację między węzłami



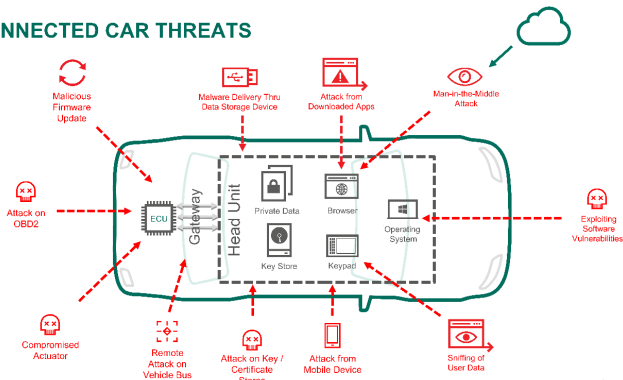
Po każdym 5 takich samych bitach automatycznie wtawiany jest bit przeciwny:

- by utrzymać synchronizację między węzłami
- nie dotyczy pól CRC, ACK

Po każdym 5 takich samych bitach automatycznie wtawiany jest bit przeciwny:

- by utrzymać synchronizację między węzłami
- nie dotyczy pól CRC, ACK
- 6 taki sam bit → błąd aktywnej stacji

## CONNECTED CAR THREATS

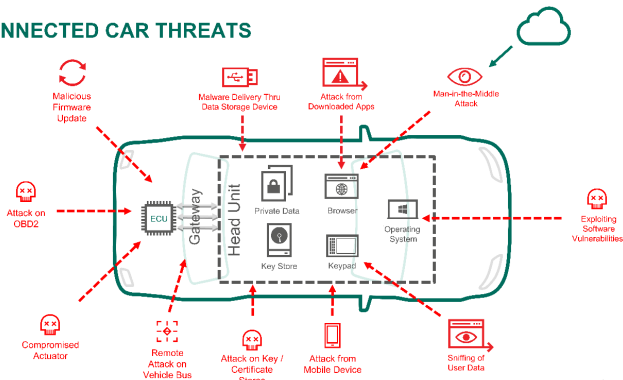


KASPERSKY

<https://www.itnewsbuzz.com/> ↔

↔ /nmw2017-kaspersky-avl-software-functions-gmbh-pave-way-for-secure-by-design-connected-cars/

## CONNECTED CAR THREATS



KASPERSKY

<https://www.itnewsbuzz.com/> ↔

↔ [/nmw2017-kaspersky-avl-software-functions-gmbh-pave-way-for-secure-by-design-connected-cars/  
www.kaspersky.com/blog/connected-car-apps-revisited/18548/](https://www.kaspersky.com/blog/connected-car-apps-revisited/18548/)