

Sterowniki Robotów

Interfejsy komunikacyjne

Wojciech Domski

Katedra Cybernetyki i Robotyki,
Politechnika Wroclawska



Wrocław University
of Science and Technology



Plan prezentacji

- 1 Wprowadzenie
- 2 Serial Peripheral Interface
- 3 Inter-Integrated Circuit
- 4 Controller Area Network
- 5 Universal Serial Bus
- 6 Universal Synchronous Asynchronous Receiver Transmitter
- 7 1-Wire



Przykładowe peryferia

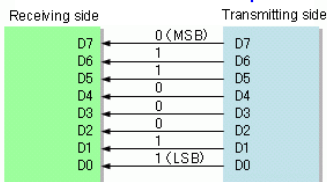
Lista najpopularniejszych peryferiów komunikacyjnych w mikrokontrolerach:

- SPI,
- I2C,
- CAN,
- SDIO,
- USB,
- USART/UART,
- FSMC.

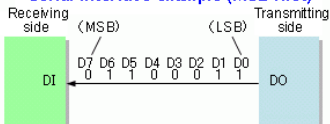


Komunikacja szeregową, a komunikacja równoległą

Parallel interface example



Serial interface example (MSB first)



1

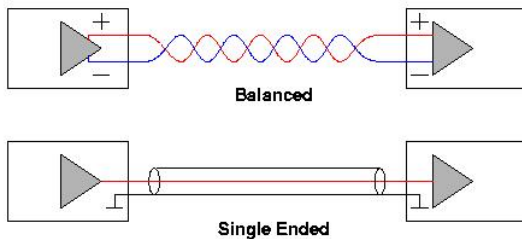


Wrocław University
of Science and Technology

¹www.wikiwand.com



Transmisja różnicowa, a transmisja z odniesieniem (1/2)



2

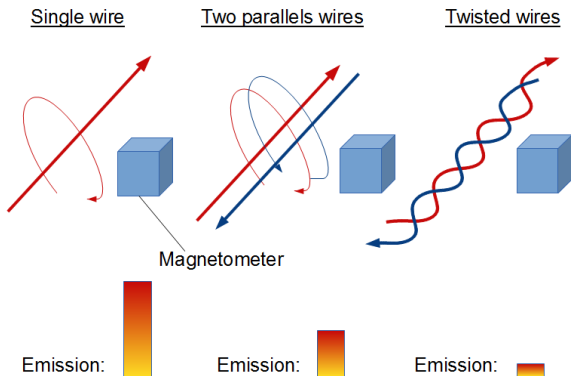


Transmisja różnicowa, a transmisja z odniesieniem (2/2)

- podwojone napięcie pary różnicowej w stosunku do nominalnego napięcia w przypadku transmisji z odniesieniem,
- odporność na zakłócenia,
- zwielokrotniona długość przewodów,
- dla wyższych częstotliwości impedancja wyjściowa wzmacniacza wyjściowego może się zmienić, powodując niewielką asymetrię. Dla pary różnicowej i identycznych wzmacniaczach, taka zmiana będzie identyczna dla obu linii, a w efekcie zostanie zniwelowane.



Zakłócenia (1/3)



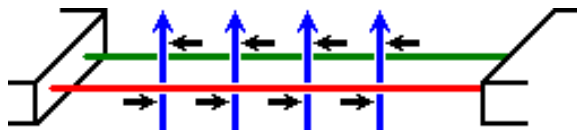
3



Wrocław University
of Science and Technology



Zakłócenia (2/3)



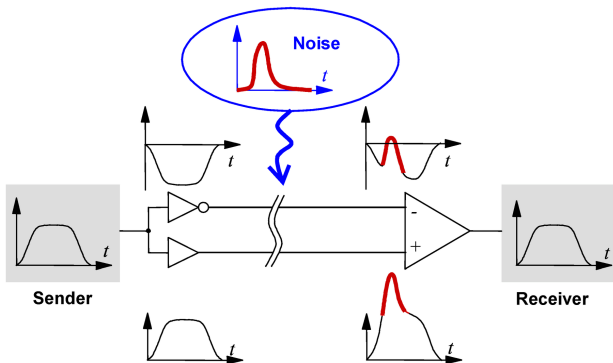
Straight cable



Twisted pair cable

**Magnetic field****Induced noise current**

Zakłócenia (3/3)



5



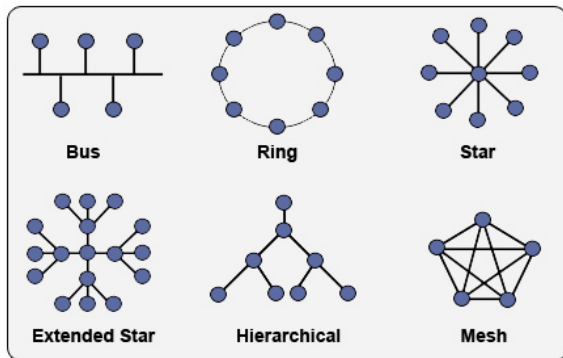
3 sbc-systems.com
 Wrocław University
 of Science and Technology
 4 root.cz

5 wikipedia.org



Wprowadzenie

Topologie sieci



6



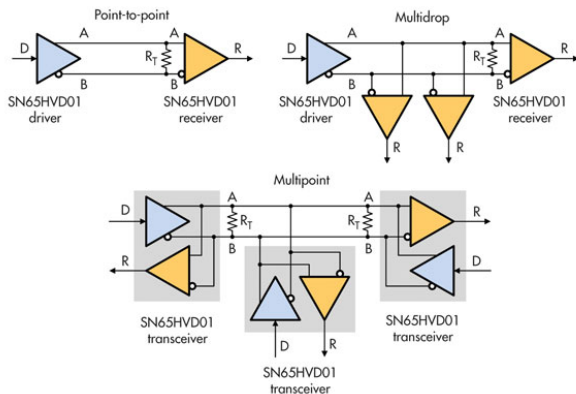
Wrocław University
of Science and Technology

⁶www.sudarshansoftech.com



Wprowadzenie

Topologie magistrali



SPI to interfejs szeregowy, który pozwala na komunikację synchroniczną zazwyczaj w obrębie danego obwodu drukowanego, lokalnie.

Komunikacja odbywa się najczęściej w konfiguracji 1-n.

Możliwe jest występowanie wielu układów nadrzędnych.

Wykorzystywany głównie do komunikacji pomiędzy mikrokontrolerem, a czujnikami, bądź układami wykonawczymi.

Pozwala on na szybką wymianę danych > 1 Mbps.

Ramka danych jest dowolnej wielkości, ale standardowo określana jest ona na długość 8 bitów (1 bajt).

Kolejność przesyłanych bitów może być ustawiona na MSB (najczęściej) lub LSB.



Linie komunikacyjne

- MOSI – Master Output Slave Input, dane przesyłane w kierunku od mastera do slave'a,
- MISO – Master Input Slave Output, dane przesyłane w kierunku do slave'a od mastera,
- SCK – sygnał zegara generowany przez mastera,
- NSS – wybór układu podrzędnego, aktywny w stanie niskim.



Tryby komunikacji

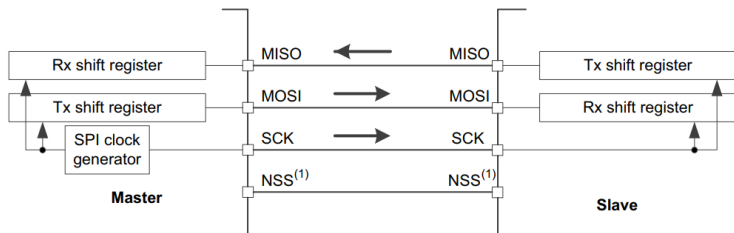
SPI może pracować w jednym z następujących trybów:

- Full-duplex,
- Half-duplex,
- Simplex.



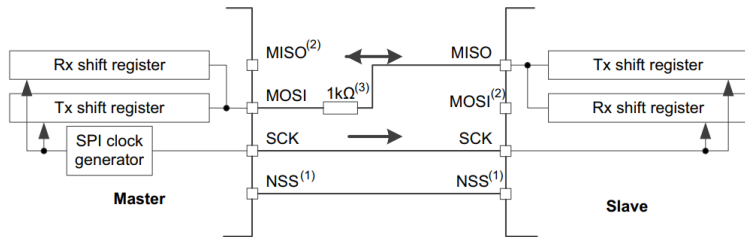
Full-duplex

Full-duplex (4 przewody) - w tym trybie odbywa się jednoczesna dwustronna komunikacja odczytu/zapisu pomiędzy masterem i slawem. Master zawsze rozpoczyna transmisję.



Half-duplex

Half-duplex (3 przewody) — połączenie skrzyżowane pomiędzy MOSI i MISO są obecne. Umożliwia dwustronną komunikację ale używana jest pojedyncza linia, zatem tylko do odczytu lub zapisu możliwa jest operacja.

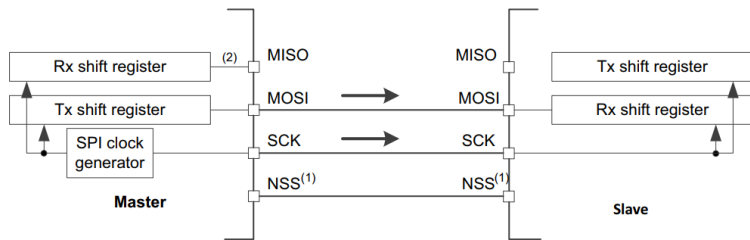


9

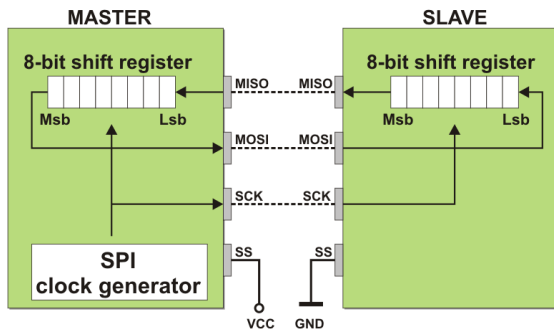


Simplex

Simplex – możliwa tylko transmisja jednokierunkowa gdzie master działa jako nadajnik, podczas gdy slave odbiera dane.



Budowa interfejsu



11

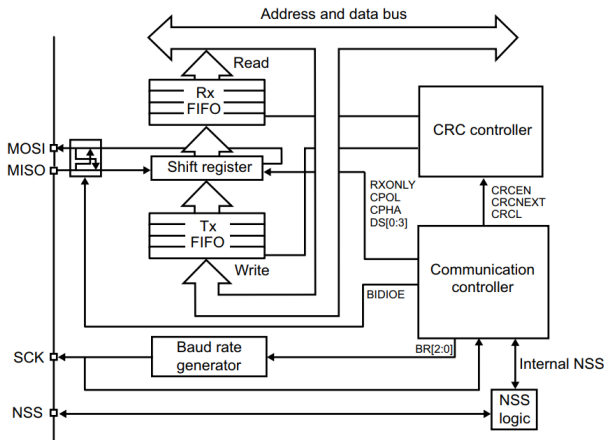


Wrocław University
of Science and Technology

¹¹learn.mikroe.com



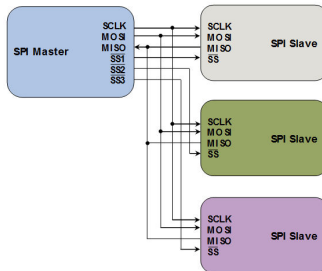
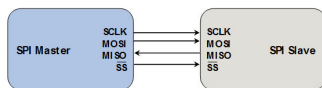
SPI w STM32



12



Topologia



13



Wrocław University
of Science and Technology

13 www.byteparadigm.com



Polaryzacja i faza zegara (1/3)

W zależności od konfiguracji polaryzacji i fazy sygnału taktującego (zegara) można wyróżnić cztery różne tryby konfiguracji.

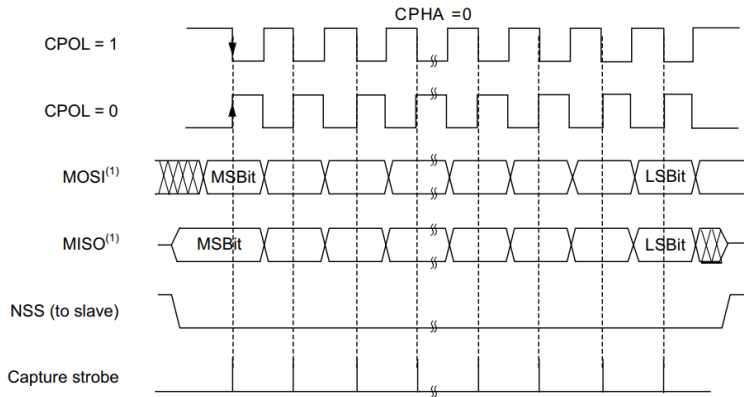
		CPHA	
		0	0
CPOL	0	00	01
	1	10	11

Polaryzacja zegara określa w jakim stanie jest linia sygnałowa, gdy transmisja nie występuje.

Faza zegara określa, na którym zboczach dane na liniach MOSI/MISO powinny być próbkowane. Innymi słowy faza określa, czy dane powinny być zatrzymywane na pierwszym, czy na drugim zboczach sygnału zegarowego



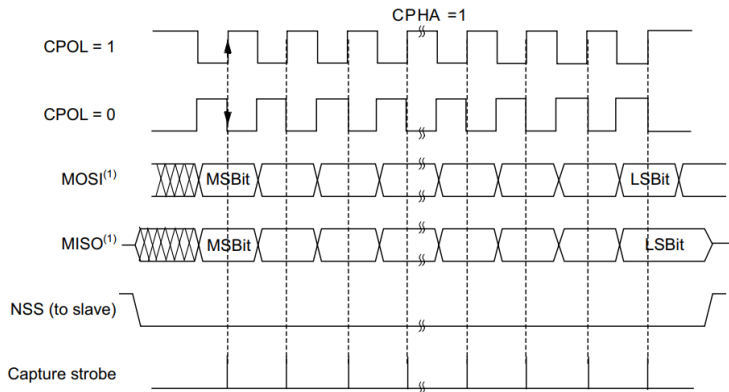
Polaryzacja i faza zegara (2/3)



14



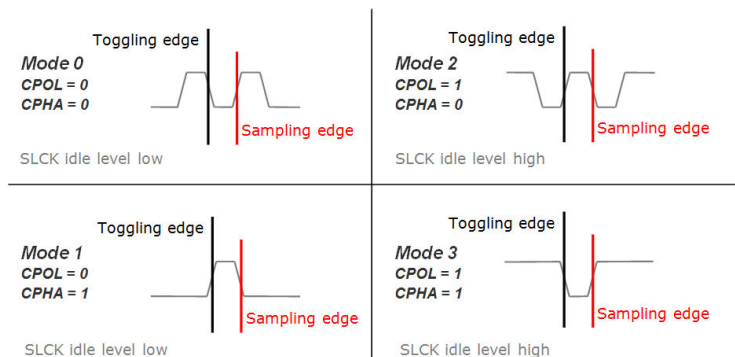
Polaryzacja i faza zegara (3/3)



15



Konfiguracja



16

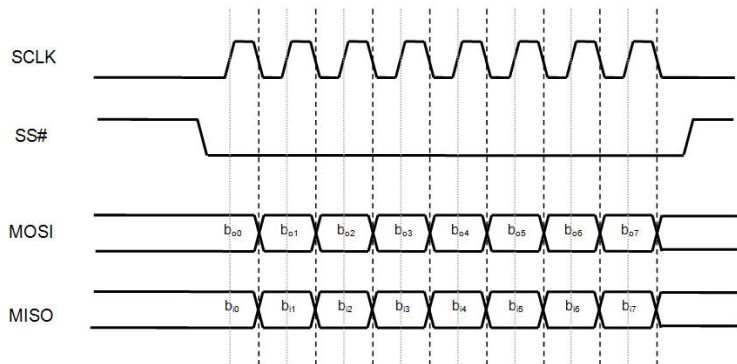


Wrocław University
of Science and Technology

16 www.byteparadigm.com



Przebiegi czasowe



17



Wrocław University
of Science and Technology

17 www.byteparadigm.com



API (1/2)

1 HAL_SPI_Transmit
2 HAL_SPI_Receive
3 HAL_SPI_TransmitReceive
4 HAL_SPI_Transmit_IT
5 HAL_SPI_Receive_IT
6 HAL_SPI_TransmitReceive_IT
7 HAL_SPI_Transmit_DMA
8 HAL_SPI_Receive_DMA
9 HAL_SPI_TransmitReceive_DMA
10
11 HAL_SPI_Abort
12 HAL_SPI_Abort_IT
13
14 HAL_SPI_DMAPause
15 HAL_SPI_DMAResume
16 HAL_SPI_DMAStop
17
18 HAL_SPI_IRQHandler
19 HAL_SPI_TxCpltCallback
20 HAL_SPI_RxCpltCallback
21 HAL_SPI_TxRxCpltCallback
22 HAL_SPI_TxHalfCpltCallback



API (2/2)

```
23 HAL_SPI_RxHalfCpltCallback
24 HAL_SPI_TxRxHalfCpltCallback
25 HAL_SPI_ErrorCallback
26 HAL_SPI_AbortCpltCallback
27
28 HAL_SPI_GetState
29 HAL_SPI_GetError
30
31 __HAL_SPI_ENABLE
32 __HAL_SPI_DISABLE
33 __HAL_SPI_ENABLE_IT
34 __HAL_SPI_DISABLE_IT
35 __HAL_SPI_GET_FLAG
36 __HAL_SPI_GET_IT_SOURCE
```



I2C (1/3)

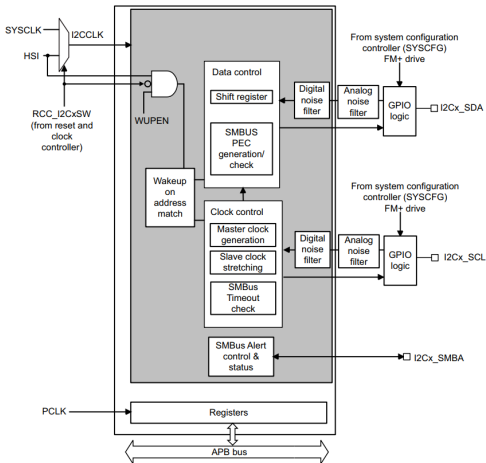
I2C to synchroniczny interfejs komunikacji szeregowej. Pozwala on na łączenie wielu urządzeń w obrębie jednego systemu. Dodatkowo posiada on możliwość konfiguracji w trybie multimaster.

Urządzenie komunikujące się po magistrali I2C może być w jednym z czterech stanów:

- Slave transmitter,
- Slave receiver,
- Master transmitter,
- Master receiver.



I2C (2/3)



18



Interfejs posiada dwie linie:

- SDA (data line) – wykorzystywana do transmisji danych,
- SCL (clock line) – linia sygnałowa generowana przez urządzenie nadrzędne.

Komunikacja odbywa się w trybie Half-Duplex.

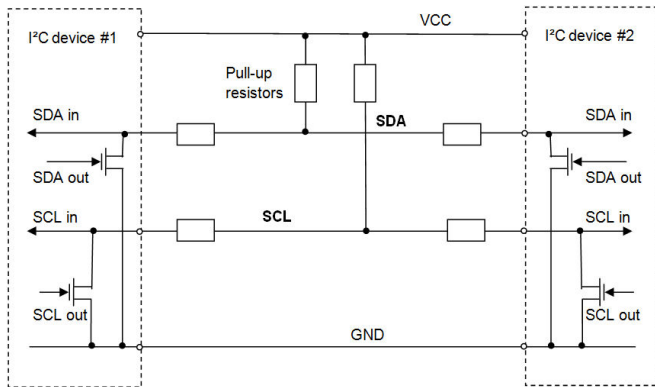
Pole danych ma rozmiar 8 bitów.

Prędkość transmisji określają trzy standardy:

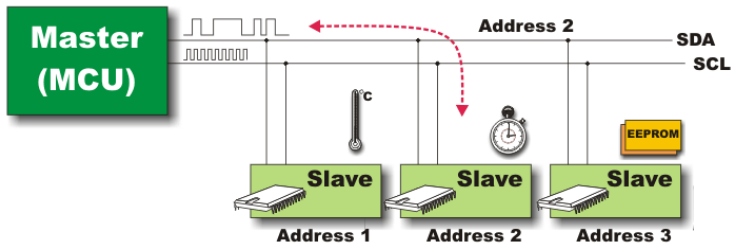
- 100kbps *Standard*,
- 400kbps *Fast*,
- 1Mbps *High Speed, Fast+*.



Budowa interfejsu



Topologia



19

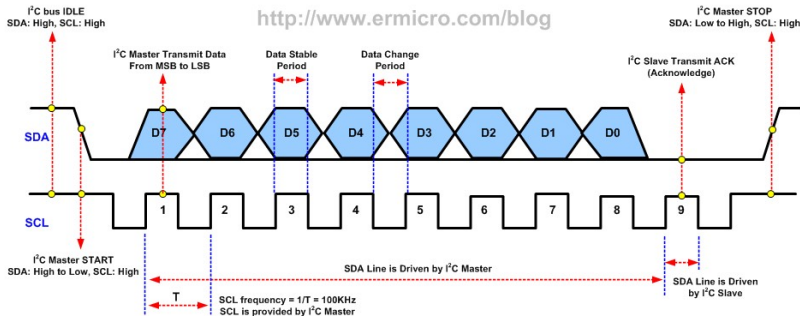


Wrocław University
of Science and Technology

¹⁹learn.mikroe.com



Przebiegi czasowe



I²C Bus Master and Slave Timing Diagram

20



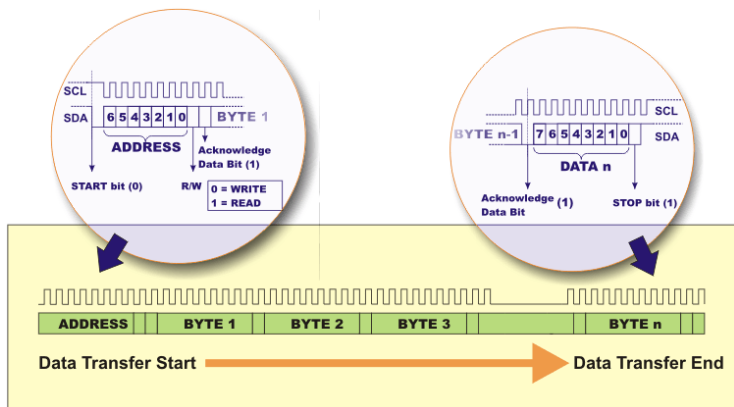
Wrocław University
of Science and Technology

20

www.ermicro.com



Protokół



21



Wrocław University
of Science and Technology

²¹learn.mikroe.com



Ramka

START	Slave address	Rd/nWr	ACK	Data	ACK	Data	ACK	STOP
1 bit	7 bits	1 bit	1 bit	8 bits	1 bit	8 bits	1 bit	1 bit

Zapis dwóch bajtów

START	Slave address	0	0	Data	0	Data	0	STOP
1 bit	7 bits	1 bit	1 bit	8 bits	1 bit	8 bits	1 bit	1 bit

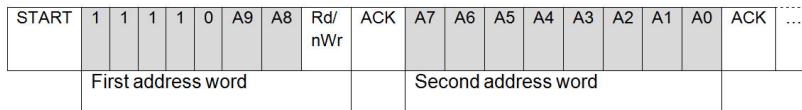
Odczyt dwóch bajtów

START	Slave address	1	0	Data	0	Data	1	STOP
1 bit	7 bits	1 bit	1 bit	8 bits	1 bit	8 bits	1 bit	1 bit

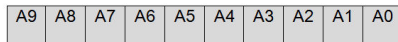
22



Adresowanie 10b



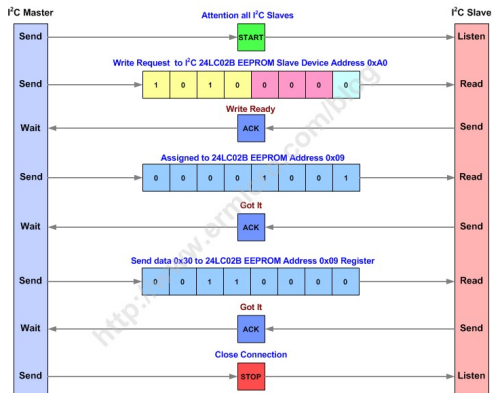
10 bits address:



23



Przykład: zapis



I²C Master Device and I²C Slave (24LC02B) Device Write Operation Handshake Diagram

24



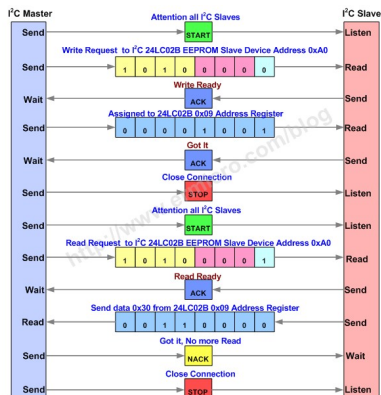
Wrocław University
of Science and Technology

24

www.ermicro.com



Przykład: odczyt

I²C Master Device and I²C Slave (24LC02B) Device Read Operation Handshake Diagram

25

Wrocław University
of Science and Technology

25

www.ermicro.com



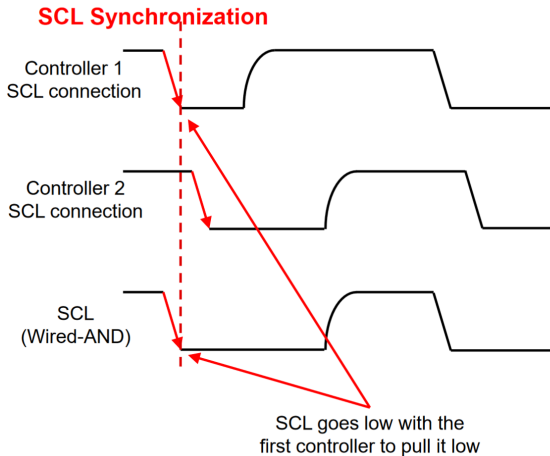
Arbitraż (1/6)

Interfejs I2C wyposażony jest w mechanizm arbitrażu. W przypadku występowania dwóch urządzeń nadrzędnych może dojść do sytuacji, w której dwa urządzenia rozpoczynają transmisję. Sytuacja taka jest nietypowa i prowadzi do błędów transmisji.

Aby wykonać arbitraż niezbędne jest monitorowanie dwóch linii SDA i SCL jednocześnie.



Arbitraż (2/6)



Arbitraż (3/6)

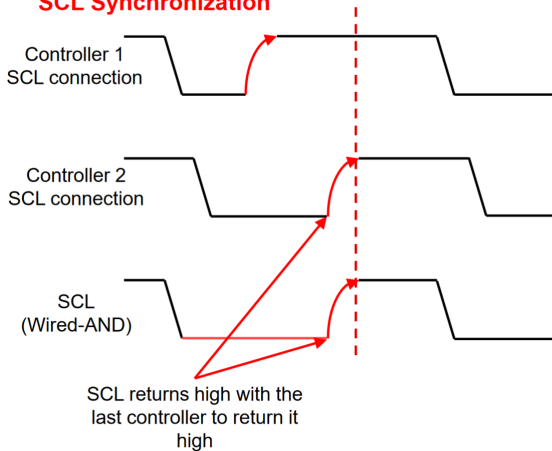
Truth Table	Controller 1	Controller 2	Resulting
	SCL	SCL	SCL
	0	0	0
	0	1	0
	1	0	0
	1	1	1

1



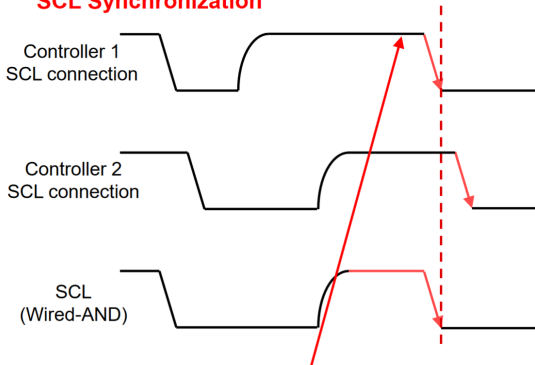
Arbitraż (4/6)

SCL Synchronization



Arbitraż (5/6)

SCL Synchronization

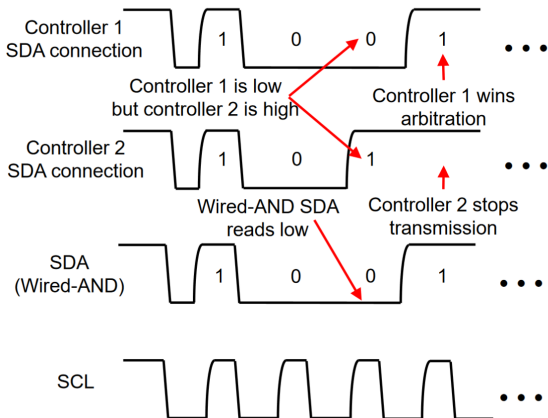


Controllers must monitor
SCL, and must keep SCL
high until all other
controllers have released
SCL high



Arbitraż (6/6)

SDA Arbitration



API (1/2)

1 HAL_I2C_Master_Transmit
2 HAL_I2C_Master_Receive
3 HAL_I2C_Slave_Transmit
4 HAL_I2C_Slave_Receive
5 HAL_I2C_Master_Transmit_IT
6 HAL_I2C_Master_Receive_IT
7 HAL_I2C_Slave_Transmit_IT
8 HAL_I2C_Slave_Receive_IT
9
10 HAL_I2C_Master_Transmit_DMA
11 HAL_I2C_Master_Receive_DMA
12 HAL_I2C_Slave_Transmit_DMA
13 HAL_I2C_Slave_Receive_DMA
14
15 HAL_I2C_Mem_Write
16 HAL_I2C_Mem_Read
17 HAL_I2C_Mem_Write_IT
18 HAL_I2C_Mem_Read_IT
19 HAL_I2C_Mem_Write_DMA
20 HAL_I2C_Mem_Read_DMA
21 HAL_I2C_IsDeviceReady



API (2/2)

```
23  __HAL_I2C_ENABLE
24  __HAL_I2C_DISABLE
25  __HAL_I2C_ENABLE_IT
26  __HAL_I2C_DISABLE_IT
27  __HAL_I2C_GET_FLAG
28  __HAL_I2C_GET_IT_SOURCE
29  __HAL_I2C_GENERATE_NACK
```



Magistrala CAN występuje na drugiej warstwie modelu OSI. Zatem dostarcza specyfikacji nie tylko warstwy fizycznej, ale również łącza danych.

Magistrala CAN implementuje:

- unikanie kolizji danych,
- wykrywanie błędów,
- automatyczną retransmisję,
- zapewnia spójność danych wśród wszystkich węzłów w sieci.



Magistrala CAN

Magistrala CAN wspiera:

- obecność wielu układów nadrzędnych (z ang. *multimaster*),
- obecność priorytetów dla ramek danych,
- arbitraż szyny danych na podstawie priorytetów,
- wykrywanie błędów oraz przywracanie normalnego stanu szyny danych na wielu poziomach,
- synchronizację danych pomiędzy węzłami w przypadku różnych źródeł taktujących.



Warstwa fizyczna

- dwukierunkowa **asynchroniczna** transmisja danych Half-Duplex poprzez pojedynczą parę różnicową,
- zwiększona odporność na zakłócenia,
- szeroki zakres napięć pracy dla transmisji z odniesieniem do różnych potencjałów masy węzłów,
- możliwość komunikacji na duże odległości, nawet 10km.

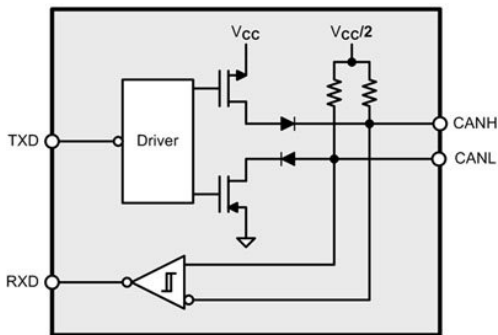


Adresacja i prędkość transmisji

- Rozmiar pola danych do 8B
- Adresowanie 11b (CAN2.0A) lub 29b (CAN2.0B)
- Szybkość transmisji: do 1Mbps (przy 40m)



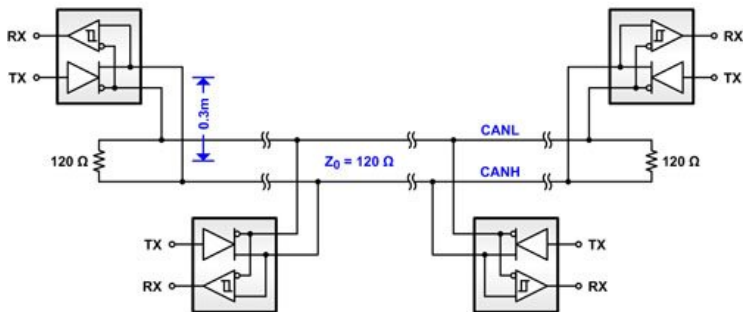
Budowa warstwy fizycznej CAN



26

Wrocław University
of Science and Technology26 www.ecnmag.com

Topologia sieci CAN (1/2)



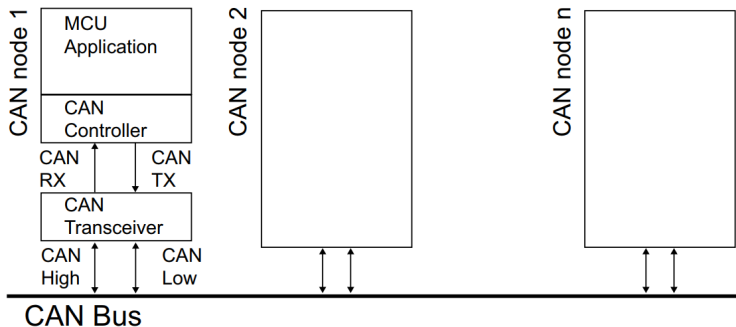
27



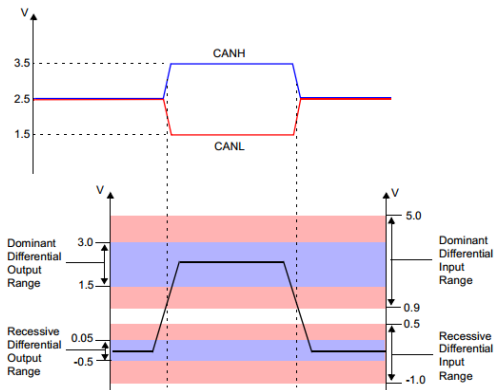
Wrocław University
of Science and Technology



Topologia sieci CAN (2/2)



Poziomy napięć w CAN



28



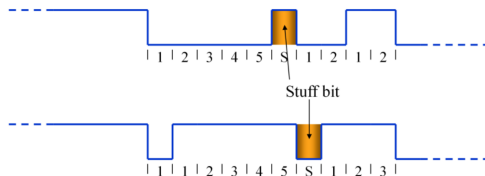
Wrocław University
of Science and Technology

²⁸Microchip



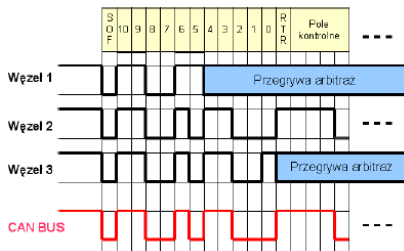
Stany magistrali i kodowanie w CAN

- Stany magistrali
 - *Recessive* (R) – logiczne 1 (stan spoczynkowy magistrali)
 - *Dominant* (D) – logiczne 0 (węzeł steruje magistralą)
- Synchronizacja zegarów: *SOF*, *Recessive-to-Dominant*
- Kodowanie NRZ (*Non Return to Zero*)
- *Bit stuffing* – po 5 identycznych stanach logicznych wstawienie przeciwnego



Protokół dostępu do medium w CAN

- *CSMA/CD-CR* Carrier Sense Multiple Access and Collision Detection with Collision Resolution
- Arbitraż



30



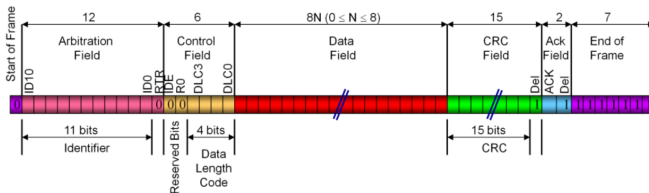
Wrocław University
of Science and Technology

³⁰canbus.pl

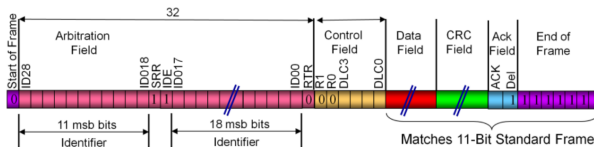


Ramki danych i RTR w CAN

Ramka danych CAN 2.0A

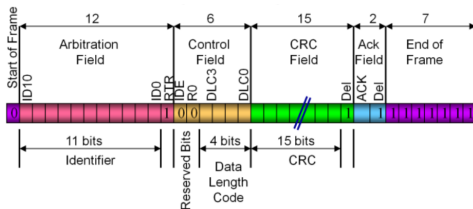


Ramka danych CAN 2.0B

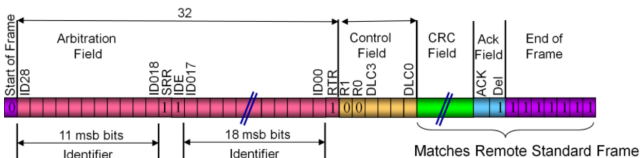


Ramki RTR i błędu oraz przeciążenia w CAN (1/2)

Ramka RTR CAN 2.0A

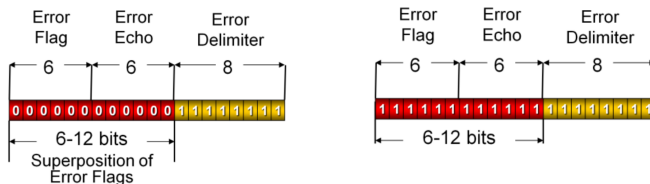


Ramka RTR CAN 2.0B

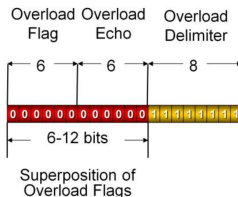


Ramki RTR i błędu oraz przeciążenia w CAN (2/2)

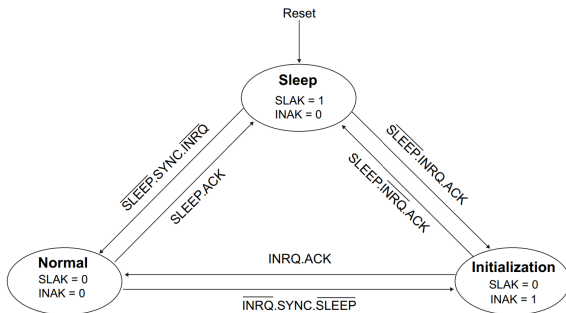
Ramka błędu (*active* po lewej, *pasive* po prawej)



Ramka przeciążenia (*overload*)



Maszyna stan węzła w sieci CAN



34

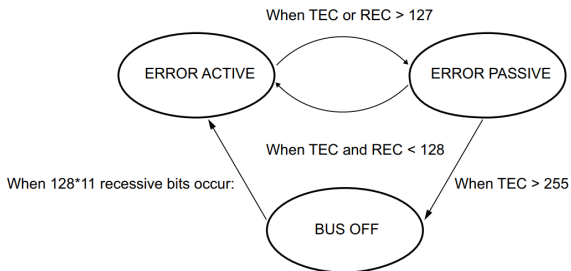


Wrocław University
of Science and Technology

34 www.st.com



Maszyna stanów błędów węzła w sieci CAN



35



Rodzaje błędów transmisji w CAN

- Bitowy - wysłanie bitu dominującego, odbiór bitu recesywnego, wyjątek to sekwencja arbitrażu
- Bit stuffing - odbiór 6 bitów o takim samym poziomie logicznym
- Format ramki - bity o ustalonych wartościach w ramce mają stan przeciwny
- Sumy kontrolnej - niezgodna suma kontrolna ramki
- Potwierdzenia - brak potwierdzenia przez odbiornik odbioru ramki (bit ACK w stanie R)



API (1/2)

1 HAL_CAN_Start
2 HAL_CAN_Stop
3
4 HAL_CAN_RequestSleep
5 HAL_CAN_WakeUp
6 HAL_CAN_IsSleepActive
7
8 HAL_CAN_AddTxMessage
9 HAL_CAN_AbortTxRequest
10 HAL_CAN_GetTxMailboxesFreeLevel
11 HAL_CAN_IsTxMessagePending
12
13 HAL_CAN_GetTxTimestamp
14 HAL_CAN_GetRxMessage
15 HAL_CAN_GetRxFifoFillLevel
16
17 HAL_CAN_ActivateNotification
18 HAL_CAN_DeactivateNotification
19 HAL_CAN_IRQHandler
20
21 HAL_CAN_GetState
22 HAL_CAN_GetError



API (2/2)

```
23 HAL_CAN_ResetError
24
25 __HAL_CAN_ENABLE_IT
26 __HAL_CAN_DISABLE_IT
27
28 __HAL_CAN_GET_FLAG
29 __HAL_CAN_CLEAR_FLAG
```



Interfejs USB (1/2)

Usb zapewnia transmisje asynchroniczną, Half-duplex typu master-slave.

Transmisja odbywa się na pojedynczej parze różnicowej.

Wyróżnia się dwie linie danych $D+$ i $D-$.

Rozmiar pola danych zmienny, maksymalnie 1024B.

Adresowanie urządzeń przy użyciu 7b adresu, a zatem możliwe jest podłączenie maksymalnie 127 urządzeń.



Interfejs USB (2/2)

W zależności od standardu można wyróżnić kilka kategorii szybkości transmisji:

- 1.5Mbps *Low Speed*, USB 1.0,
- 12Mbps *Full Speed*, USB 1.0,
- 480Mbps *High Speed*, USB 2.0,
- 5Gbps *Super Speed*, USB 3.0,
- 10Gbps *Super Speed+*, USB 3.1,
- 20Gbps *Super Speed+*, USB 3.2.

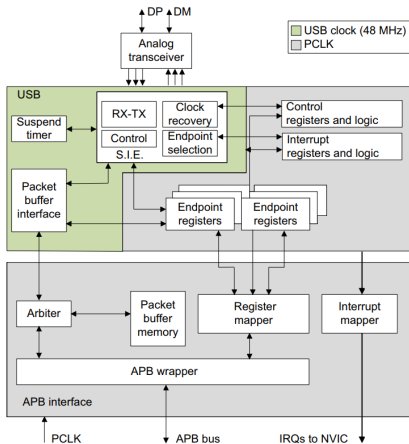
Maksymalna długość interfejsu do 5 metrów.

Urządzenie może działać w trybie:

- Plug-and-Play,
- Hot Swapping,
- Power Delivery, zasilania.



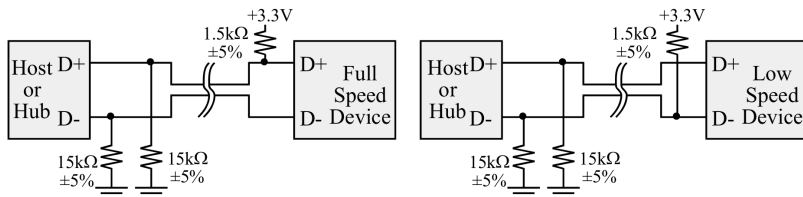
USB



36



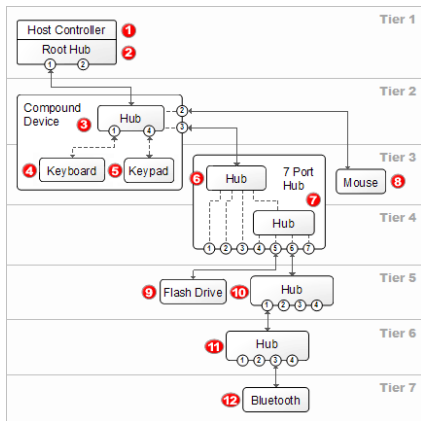
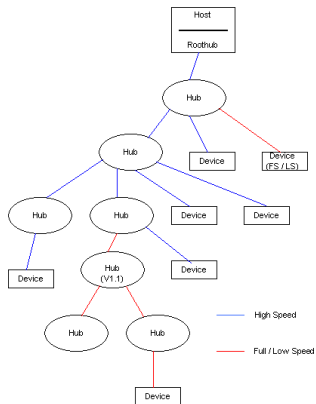
Detekcja LS, HS



37



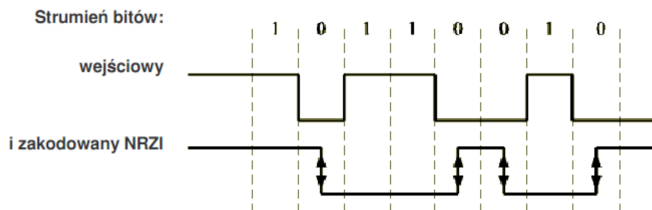
Topologia (Tiered star topology)



38

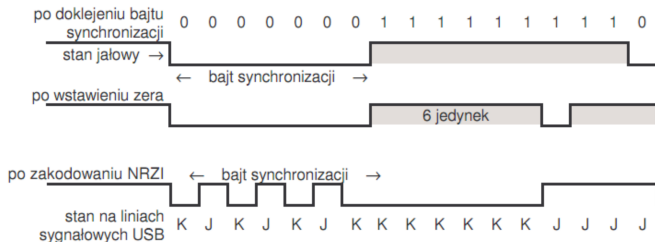

 Wrocław University
of Science and Technology

 38 www.usbmadesimple.co.uk


Kodowanie NRZI (*Non Return to Zero Invert*) (1/2)

- Kodowanie: 0 zmiana sygnału, 1 sygnał pozostaje bez zmian
- Bit stuffing – wstawianie logicznego 0 po 6 logicznych 1
- Oznaczenia: J poziom niski po zakodowaniu, K stan przeciwny



Kodowanie NRZI (*Non Return to Zero Invert*) (2/2)

39

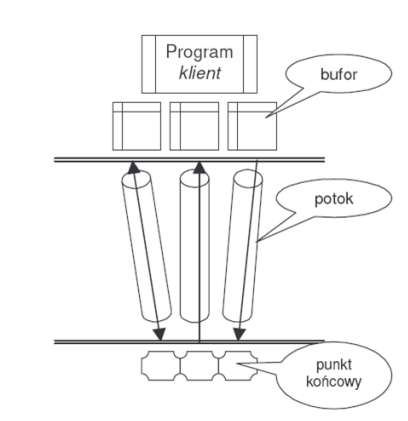


Stan linii

- Detached
- Connect
- Idle / J
- Sync
- EOP
- Reset
- Suspend
- Resume (host)
- Resume (device)
- Keep alive



Przepływ danych



40



Wrocław University
of Science and Technology

www.usb_pr.republika.pl



Przepływ danych

- Punkt końcowy – wydzielona, jednoznacznie rozpoznawalna część urządzenia, niezależnie konfigurowana, maksymalnie 15 w urządzeniu, obligatoryjny punkt 0 do konfiguracji
- Potok – abstrakcyjny obiekt, realizujący przepływ danych pomiędzy buforem w pamięci komputera, a punktem końcowym w urządzeniu, dwa rodzaje:
 - potok danych – zawartość nieistotna dla magistrali,
 - potok sterujący – ustalony format ramek



Tryby przesyłania danych

- *Control Transfers* – konfiguracja i kontrola urządzenia,
- *Bulk Data Transfers* – przesyłanie danych ze zmienną szybkością dostosowaną do aktualnie dostępnego pasma
- *Interrupt Data Transfers* – transfer niewielkich pakietów danych, z niewielkim opóźnieniem, z gwarantowanym czasem dostarczenia, bez błędów
- *Isochronous Data Transfers* – transfer w trybie RT (po negocjacji pasma i rozmiaru danych)

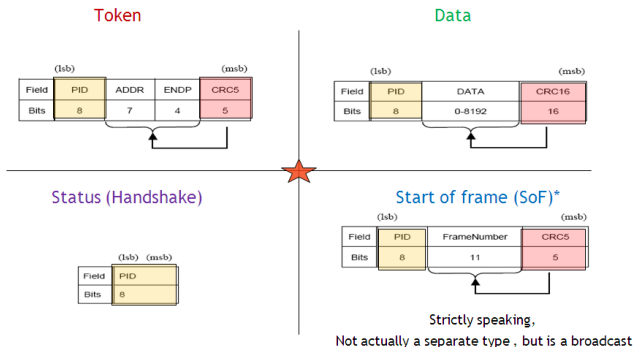


Rozmiary ramek

Type	Important attributes	Max size LS	Max size FS	Max size HS	Examples
Interrupt	Quality + time	8	64	3072	Mouse, keyboard
Bulk	Quality	-	64	512	Printer, scanner
Isochronous	time	-	1023	3072	Audio, video
Control	Quality + time	8	64	64	System control



Rodzaje ramek



41

Wrocław University
of Science and Technology41 embeddedinn.wordpress.com

Komunikacja na magistrali (1/2)

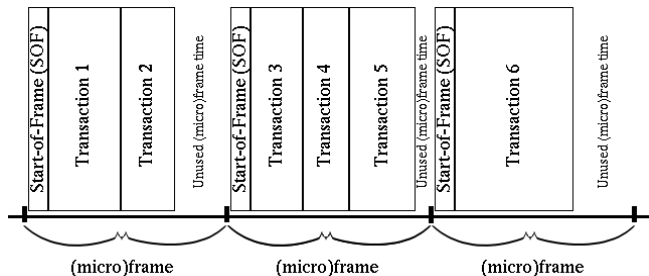
Tryby synchronizacji w przypadku stałego pasma:

- asynchroniczna - dane przesyłane zgodnie z zegarem urządzenia,
- synchroniczna - Start-of-Frame,
- adaptacyjna - synchronizacja zegara w oparciu o liczbę ramek przesłanych przez hosta.



Komunikacja na magistrali (2/2)

- SoF: 1ms LS i FS, 125 μ s HS

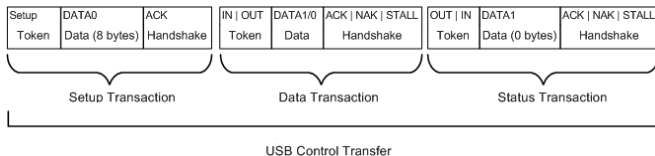


42



Transakcja

- Ramka *Token*
- Ramka *Data* (opcjonalnie)
- Ramka *Handshake*



43

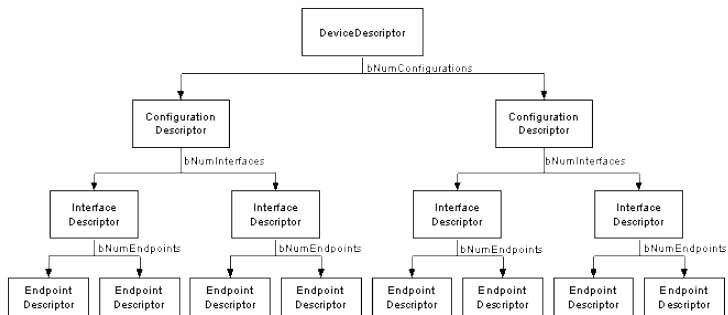


Wrocław University
of Science and Technology

⁴³msdn.microsoft.com



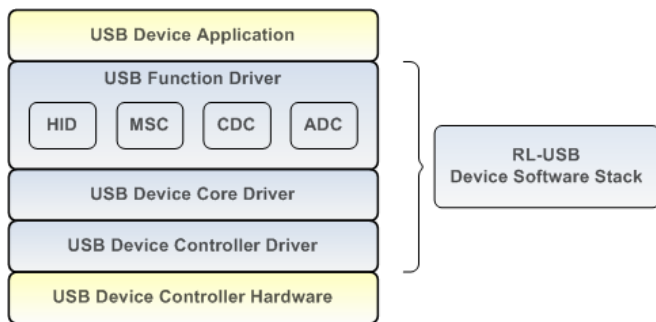
Deskryptory



44

Wrocław University
of Science and Technology44 wiki.osdev.org

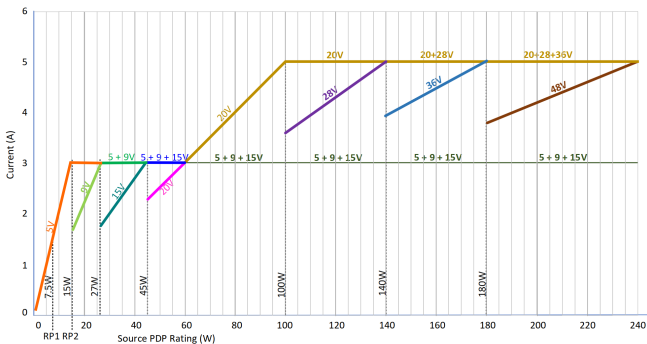
Stos interfejsu USB



45



USB Charging (Power Delivery)



46

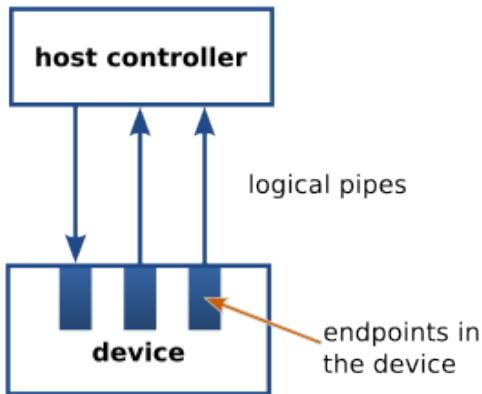


Wrocław University
of Science and Technology

46 www.usb.org



Endpoint (1/3)



47

Wrocław University
of Science and Technology

Endpoint (2/3)

Endpoint to unikalnie identyfikowany fragment urządzenia USB, który jest punktem końcowym w potoku komunikacyjnym pomiędzy urządzeniem, a hostem.
Każde urządzenie logiczne jest kompozycją niezależnych endpointów.



Endpoint (3/3)

Każde urządzenie logiczne jest wyposażone w unikalny adres nadany przez system podczas podłączania urządzenia.

Każdemu endpointowi na urządzeniu przydzielany jest unikalny identyfikator określony przez urządzenie, zwany numerem endpoint.

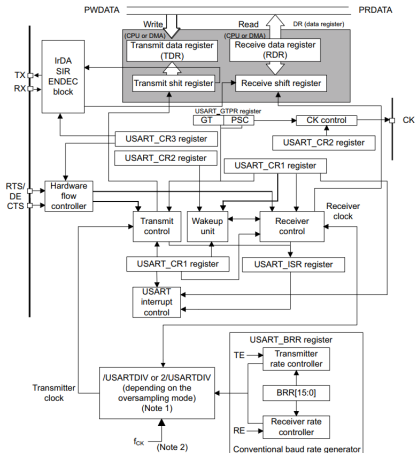
Trójka: adres urządzenia, numeru endpointa i kierunku transmisji umożliwiającą jednoznaczną identyfikację endpointa.

Każdy endpoint ma określony kierunek przepływu danych.

Każdy endpoint jest jednokierunkowym połączeniem.



USART (1/2)



48



USART (2/2)

Universal Synchronous Asynchronous Receiver Transmitter (USART) oferuje transmisję synchroniczną, bądź asynchroniczną.

Możliwy jest jednoczesny odbiór i nadawania, Half-Duplex, nadawanie lub odbiór Half-Duplex, a także nadawanie, albo odbieranie Simplex.

Rozmiar pola danych to zazwyczaj 8 bitów.

Maksymalna prędkość transmisji jaką można osiągnąć to 35Mbps w zależności od wybranej warstwy fizycznej oraz dystansu.

Możliwa jest komunikacja typu multimaster.



Linie danych RS-232 (1/2)

- **RX** – linia odbiorcza,
- **TX** – linia nadawcza,
- **CK** – linia taktująca (dla komunikacji synchronicznej),
- **CTS** – (z ang. *clear to send*), umożliwia blokowanie nadawanie po zakończeniu transferu danych, gdy w stanie wysokim,
- **RTS** – (z ang. *request to send*), aktywny w stanie niskim, gdy kontroler jest gotowy na przyjęcie danych,
- **RTR** – (z ang. *request to receive*), gotowy do odbioru,
- **DCD** – (z ang. *data carrier detected*), traw odbiór danych,
- **DTE** – (z ang. *data terminal ready*), gotowy do odbioru,
- **DSR** – (z ang. *data set ready*), gotowy do nadawania i odbioru danych,



Linie danych RS-232 (2/2)

- **RI** – (z ang. *ring indicator*), wykryto „wzwanie” na linię,
- **DE** – (z ang. *driver enabled*), sygnalizuje gotowość do nadawania w odniesieniu do urządzenia zewnętrznego.



Ramka danych

Liczba bitów	1	5-9	0-1	0-2
Rodzaj	Start Bit	Data Frame	Parity Bits	Stop Bits



Parzystość

Bit parzystości:

$$b_e = d_n \oplus \dots \oplus d_2 \oplus d_1 \oplus 0. \quad (1)$$

Bit nieparzystości:

$$b_o = d_n \oplus \dots \oplus d_2 \oplus d_1 \oplus 1. \quad (2)$$

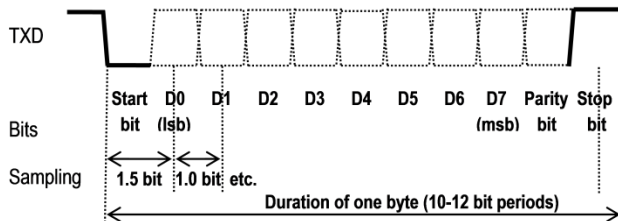


Konfiguracja

- baudrate, prędkość transmisji danych,
- rozmiar ramki danych,
- parzystość – (parzystość/nieparzystość/brak),
- liczba bitów stopu.



Przebiegi czasowe w interfejsie szeregowym



49

Wrocław University
of Science and Technology49
mbed.org

Porównanie interfejsów RS232/RS422/RS485

	RS232	RS423	RS422	RS485
Differential	no	no	yes	yes
Max number of drivers	1	1	1	32
Max number of receivers	1	10	10	32
Modes of operation	half duplex full duplex	half duplex	half duplex	half duplex
Network topology	point-to-point	multidrop	multidrop	multipoint
Max distance (acc. standard)	15 m	1200 m	1200 m	1200 m
Max speed at 12 m	20 kbs	100 kbs	10 Mbs	35 Mbs
Max speed at 1200 m	(1 kbs)	1 kbs	100 kbs	100 kbs
Max slew rate	30 V/ μ s	adjustable	n/a	n/a
Receiver input resistance	3..7 k Ω	\geq 4 k Ω	\geq 4 k Ω	\geq 12 k Ω
Driver load impedance	3..7 k Ω	\geq 450 Ω	100 Ω	54 Ω
Receiver input sensitivity	\pm 3 V	\pm 200 mV	\pm 200 mV	\pm 200 mV
Receiver input range	\pm 15 V	\pm 12 V	\pm 10 V	-7..12 V
Max driver output voltage	\pm 25 V	\pm 6 V	\pm 6 V	-7..12 V
Min driver output voltage (with load)	\pm 5 V	\pm 3.6 V	\pm 2.0 V	\pm 1.5 V

50

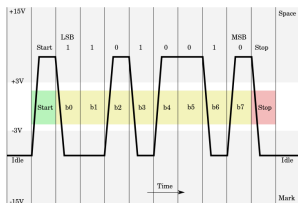
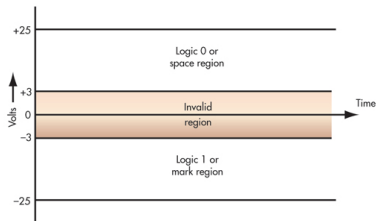
Wrocław University
of Science and Technology

50

www.lammertbies.nl



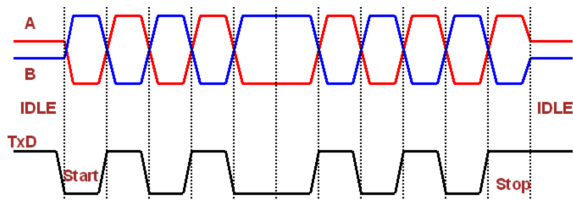
Poziomy napięć w interfejsie RS232



51



Poziomy napięć w interfejsie RS422/RS485



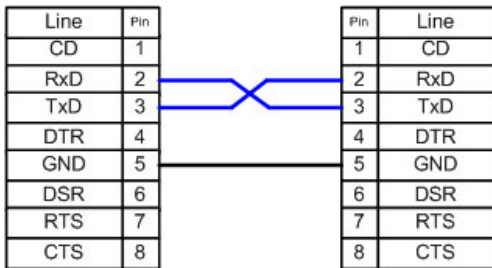
52

Wrocław University
of Science and Technology

52 Texas Instruments



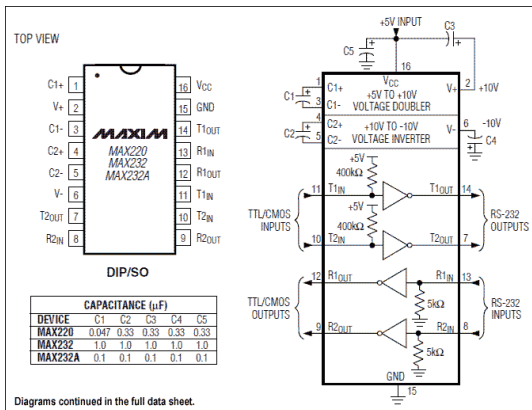
Interfejs RS232 – point-to-point, null modem



53



MAX232



54

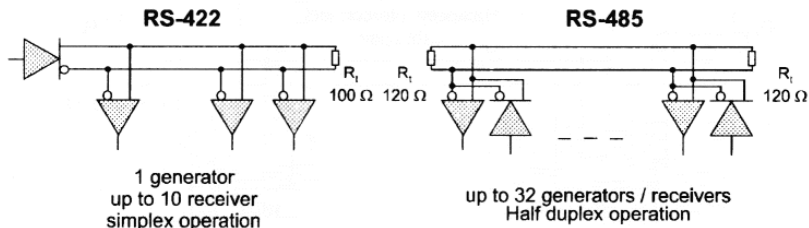

 Wrocław University
of Science and Technology

54 Maxim



Topologia RS422/RS485

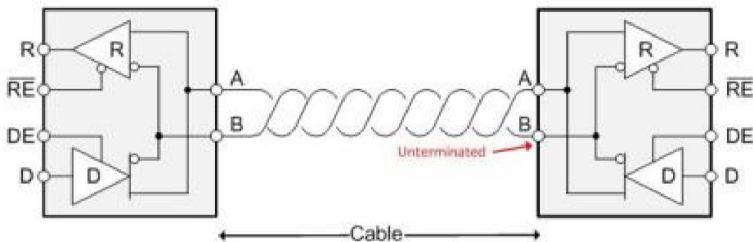
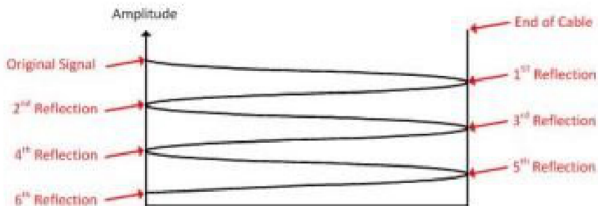
Comparison RS422 - RS485



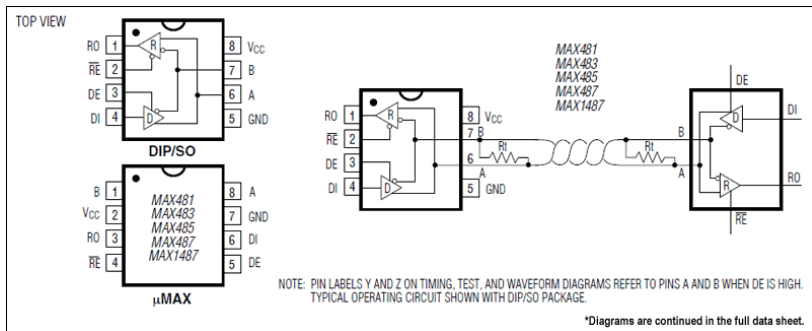
-7V to +7V	Max common mode voltage	-7 V to +12 V
4 k Ω	Receiver input impedance	12 k Ω
100 Ω	Minimum generator load	60 Ω
<150 mA to GND	Generator short circuit current	<250mA to -7 V/+12 V



Terminator, a signal obdity



MAX485



57



Wrocław University
of Science and Technology

57 Maxim



API (1/2)

1 HAL_UART_Transmit
2 HAL_UART_Receive
3 HAL_UART_Transmit_IT
4 HAL_UART_Receive_IT
5 HAL_UART_Transmit_DMA
6 HAL_UART_Receive_DMA
7
8 HAL_UART_DMAPause
9 HAL_UART_DMAResume
10 HAL_UART_DMAStop
11
12 HAL_UART_Abort
13 HAL_UART_AbortTransmit
14 HAL_UART_AbortReceive
15 HAL_UART_Abort_IT
16 HAL_UART_AbortTransmit_IT
17 HAL_UART_AbortReceive_IT
18
19 HAL_UART_IRQHandler
20 HAL_UART_TxCpltCallback
21 HAL_UART_TxHalfCpltCallback
22 HAL_UART_RxCpltCallback



API (2/2)

```
23 HAL_UART_RxHalfCpltCallback
24 HAL_UART_ErrorCallback
25 HAL_UART_AbortCpltCallback
26 HAL_UART_AbortTransmitCpltCallback
27 HAL_UART_AbortReceiveCpltCallback
28
29 __HAL_UART_ENABLE
30 __HAL_UART_DISABLE
31 __HAL_UART_ENABLE_IT
32 __HAL_UART_DISABLE_IT
33 __HAL_UART_GET_FLAG
34 __HAL_UART_GET_IT_SOURCE
```

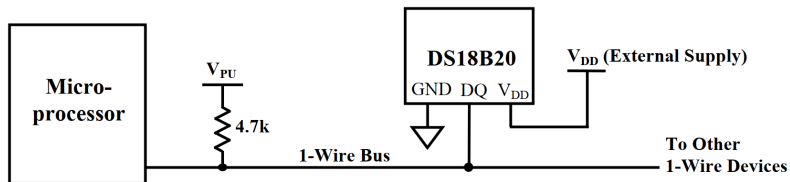


1-Wire

- pojedyncza linia danych,
- zasilanie pasożytnicze (z ang. *parasite*) lub z linii zasilającej,
- Half-duplex,
- prędkość transmisji do 22kbit/s



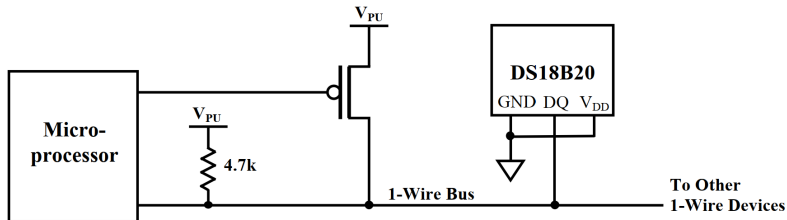
Zasilanie (1/2)



58



Zasilanie (2/2)



59

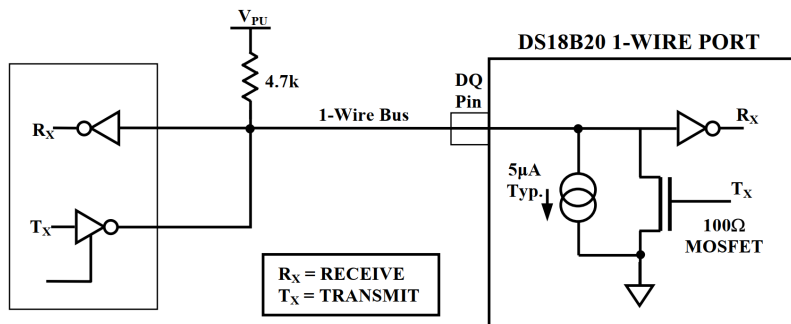


Wrocław University

of Science and Technology

<http://maximintegrated.com/>
<http://maximintegrated.com/>


Budowa interfejsu



60

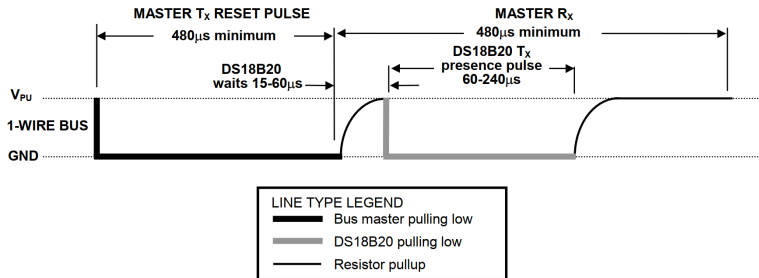


Wrocław University
of Science and Technology

⁶⁰<http://maximintegrated.com/>

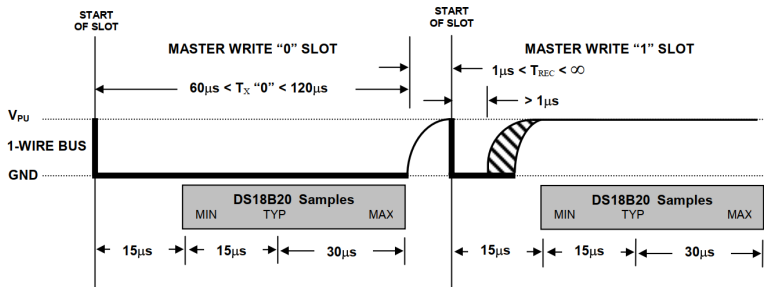


Sygnal resetu



61

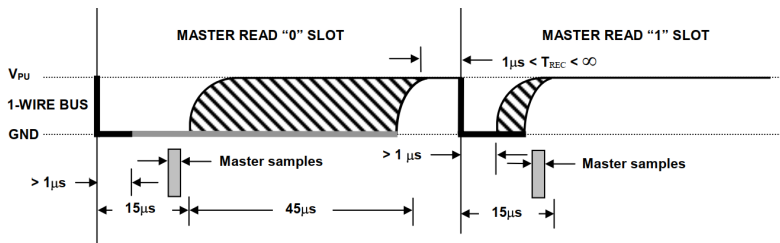




62



Odczyt



63

