

# KOMPUTEROWE SYSTEMY POMIAROWE

**Dr inż. Eligiusz PAWŁOWSKI**

**Politechnika Lubelska**

**Wydział Elektrotechniki i Informatyki**

*Prezentacja do wykładu dla EMST - ITwE*

Semestr zimowy

Wykład nr 4



# Prawo autorskie

Niniejsze materiały podlegają ochronie zgodnie z **Ustawą o prawie autorskim i prawach pokrewnych** (Dz.U. 1994 nr 24 poz. 83 z późniejszymi zmianami).

Materiał ten udostępniam **do celów dydaktycznych** jako materiały pomocnicze do wykładu z przedmiotu Komputerowe Systemy Pomiarowe prowadzonego dla studentów Wydziału Elektrotechniki i Informatyki Politechniki Lubelskiej. Mogą z nich również korzystać inne osoby zainteresowane tą tematyką. Do tego celu materiały te można **bez ograniczeń przeglądać, drukować i kopiować wyłącznie w całości**.

Wykorzystywanie tych materiałów bez zgody autora w inny sposób i do innych celów niż te, do których zostały udostępnione, **jest zabronione**.

W szczególności **niedopuszczalne jest**: usuwanie nazwiska autora, edytowanie treści, kopiowanie fragmentów i wykorzystywanie w całości lub w części do własnych publikacji.

Eligiusz Pawłowski

## Uwagi dydaktyczne

Niniejsza prezentacja stanowi **tylko i wyłącznie materiały pomocnicze** do wykładu z przedmiotu Komputerowe Systemy Pomiarowe prowadzonego dla studentów Wydziału Elektrotechniki i Informatyki Politechniki Lubelskiej. Udostępnienie studentom tej prezentacji nie zwalnia ich z konieczności sporządzania **własnych notatek z wykładów** ani też nie zastępuje **samodzielnego studiowania** obowiązujących podręczników.

Tym samym zawartość niniejszej prezentacji w szczególności **nie może być** traktowana jako zakres materiału obowiązujący na egzaminie.

Na egzaminie obowiązujący jest **zakres materiału faktycznie wyłożony podczas wykładu** oraz zawarty w odpowiadających mu fragmentach **podręczników** podanych w wykazie literatury do wykładu.

Eligiusz Pawłowski

## Podstawowe parametry systemów interfejsu

Do podstawowych parametrów **każdego systemu interfejsu** można zaliczyć:

- 1–Zasięg i szybkość transmisji,**
- 2–Struktura połączeń: logiczna i fizyczna,**
- 3–Sposób transmisji porcji informacji: bitów, bajtów i ramek,**
- 4–Zasada czasowej koordynacji transmisji (synchronizacja),**
- 5–Sposób adresowania urządzeń,**
- 6-Parametry elektryczne, mechaniczne, okablowanie.**

← Poprzedni wykład

## Struktury połączeń logicznych w systemach interfejsu

**Logiczna struktura połączeń** opisuje drogę, którą pokonują dane podczas transmisji w systemie interfejsu od jednego urządzenia do następnych, niezależnie od fizycznego sposobu realizacji połączeń (okablowania).

Wyróżniamy następujące **struktury połączeń logicznych:**

**1–radialna** (gwiazdzista, *ang. star*),

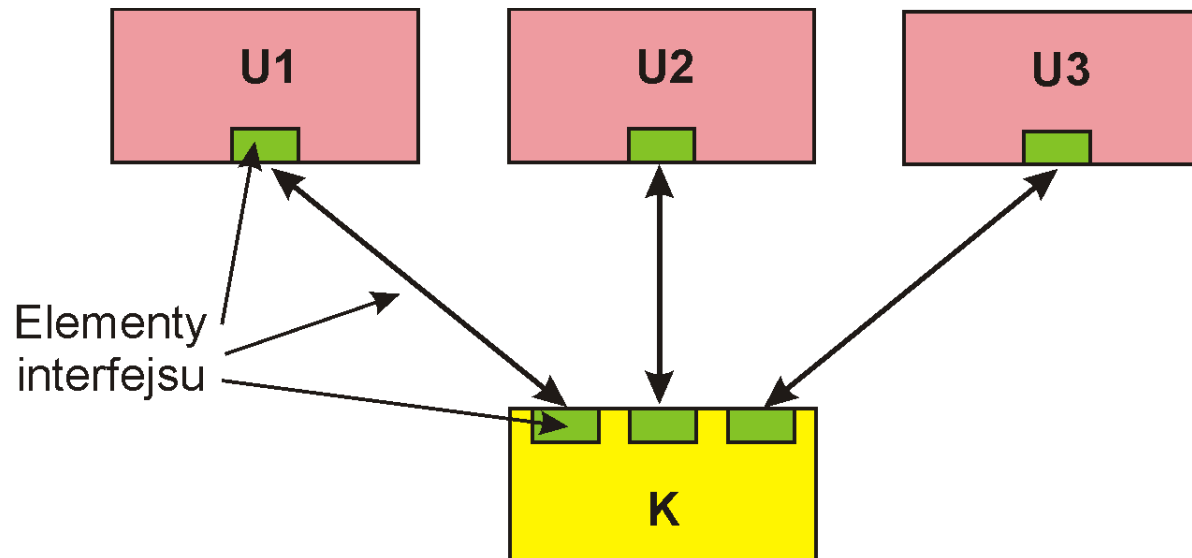
**2–drzewo**, (*ang. tree*),

**3–magistrala** (*ang. bus*),

**4–kaskadowa otwarta i zamknięta** (*ang. daisy chain*,).

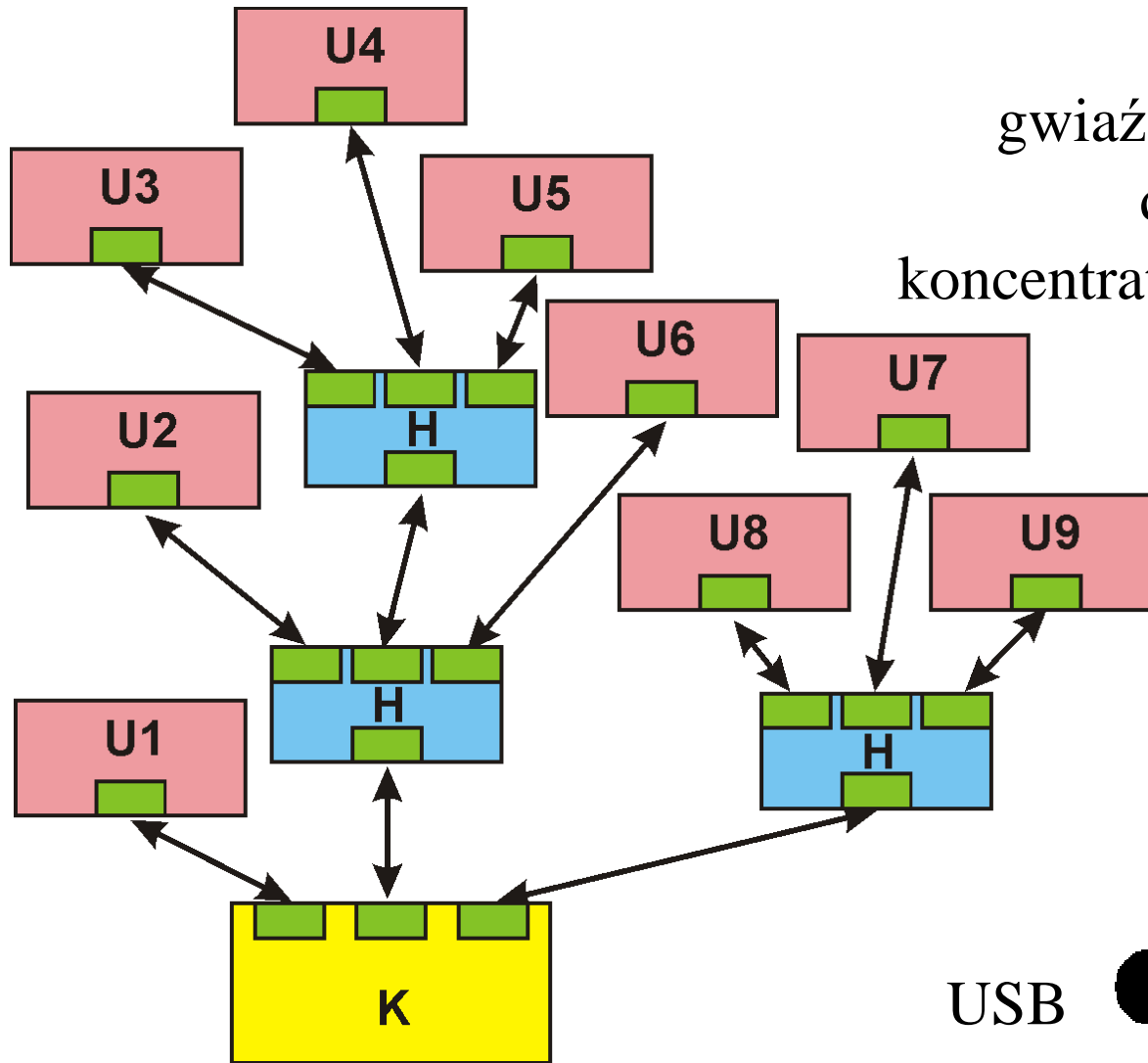
## Struktura logiczna typu radialnego, gwiazdzista (*star*)

Każde urządzenie **U1**, **U2**, **U3**, ... jest dołączone do kontrolera **K** oddzielnym interfejsem

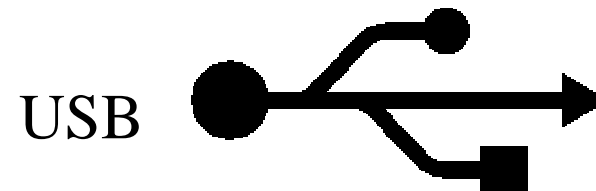


RS 232, USB, Centronics (tryb standardowy SPP)

# Struktura logiczna typu drzewo (tree)

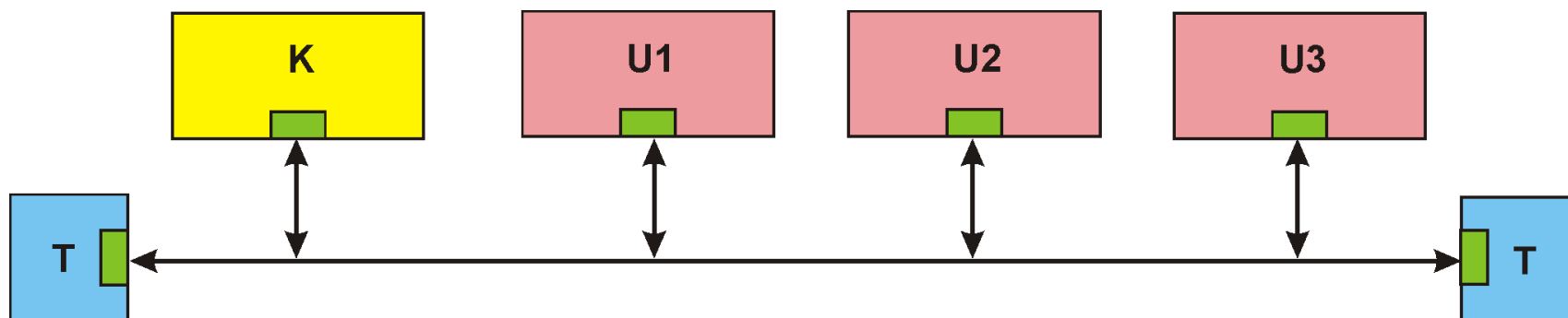


Struktura logiczna gwiazdzista rozbudowana dzięki zastosowaniu koncentratorów **H** (*ang. Hub*)



## Struktura logiczna typu magistrala (*bus*)

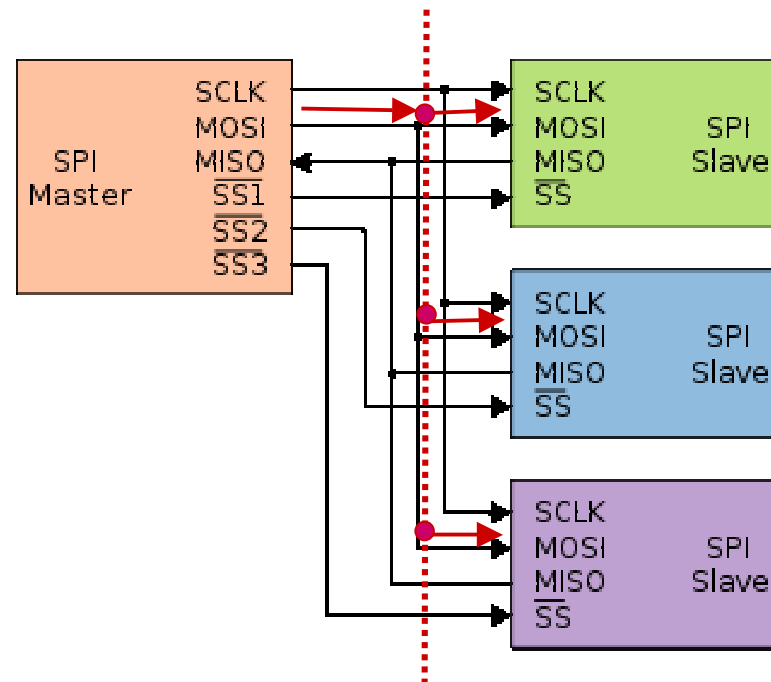
Wszystkie urządzenia **U1**, **U2**, **U3**, ... oraz kontroler **K** dołączone są poprzez układy interfejsu równoległe do magistrali, zakończonej (opcjonalnie) terminatorami **T**.



ISA, PCI, RS485, IEEE 488, SPI, I2C



## Struktura logiczna typu magistrala, przykład SPI



**SCLK** — Serial Clock (output from master)

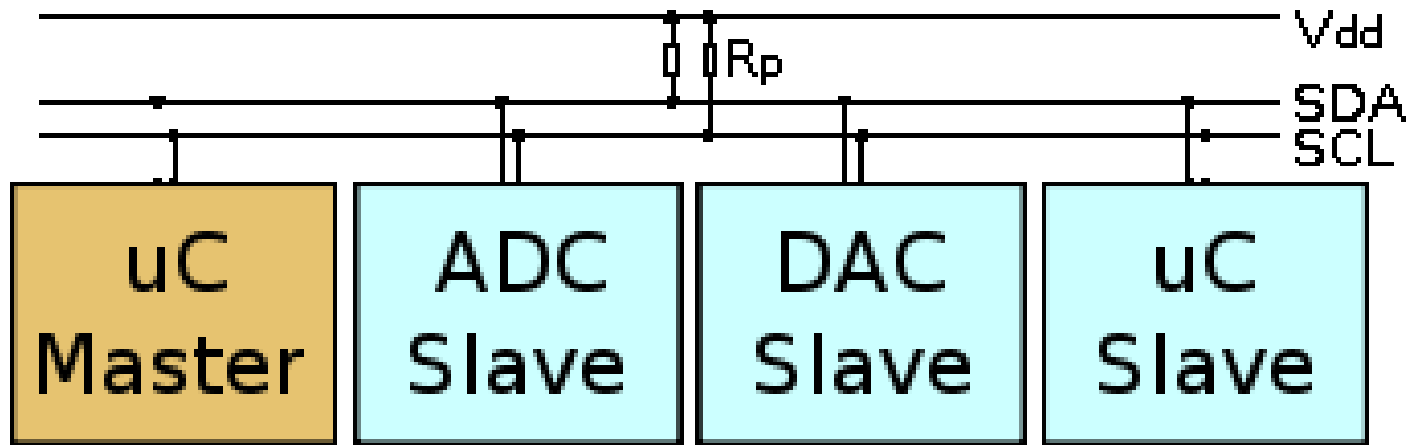
**MOSI/SIMO** — Master Output, Slave Input (output from master)

**MISO/SOMI** — Master Input, Slave Output (output from slave)

**SS** — Slave Select (active low; output from master)

**SPI** (*Serial Peripheral Interface Bus*) w konfiguracji magistrali

## Struktura logiczna typu magistrala, przykład I2C



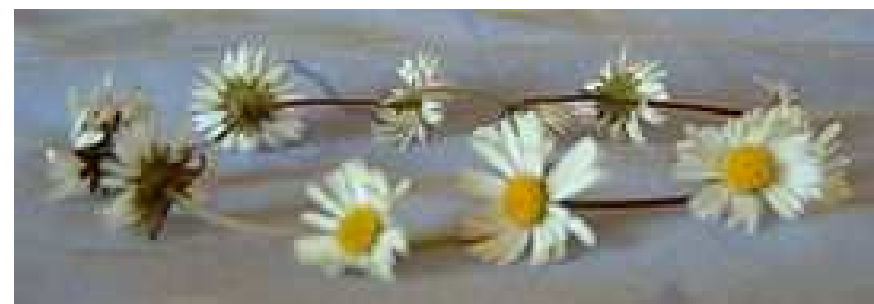
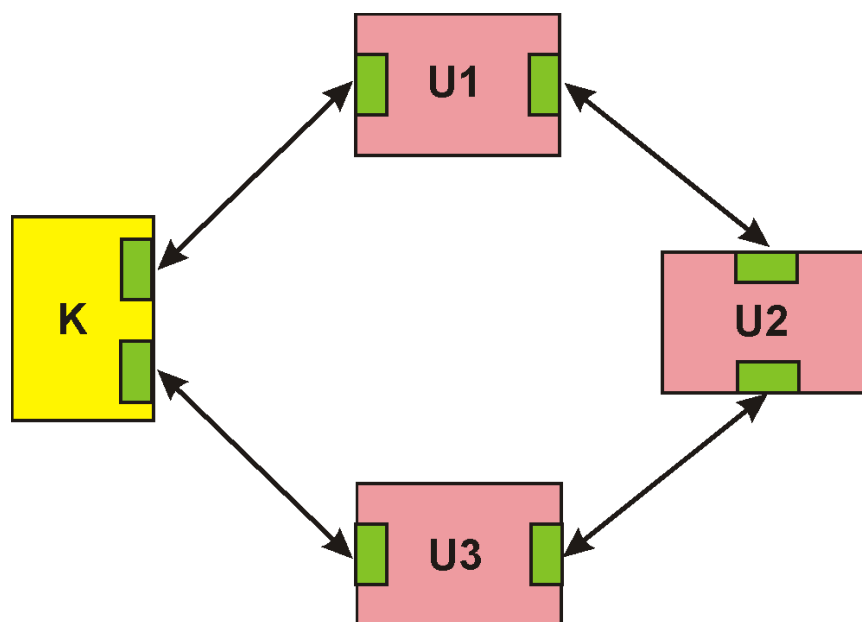
**SDA** (*Serial DATA line*) – linia do przesyłania danych

**SCL** (*Serial CLOCK line*) – linia do przesyłania sygnału zegarowego

**I2C** (*Inter-Integrated Circuit*) w konfiguracji magistrali

## Struktura logiczna kaskadowa zamknięta (*daisy chain - ring*)

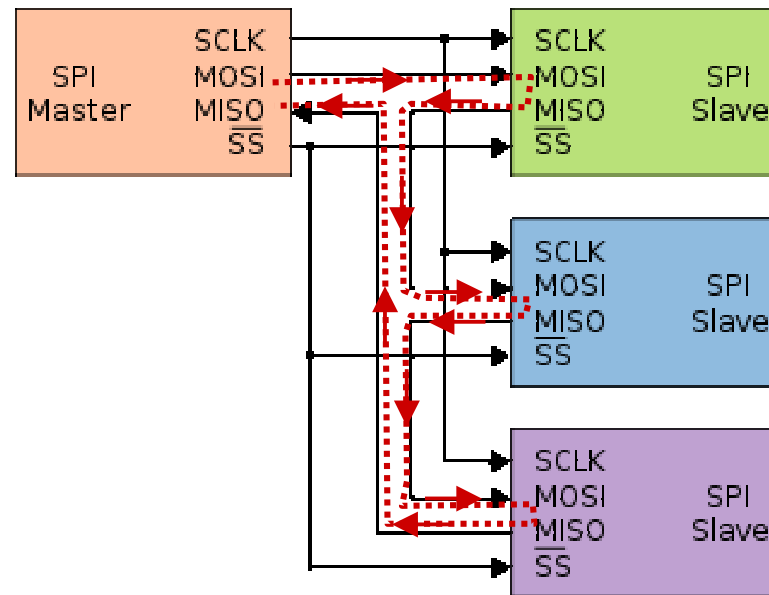
Wszystkie urządzenia **U1**, **U2**, **U3**, ... oraz kontroler **K** tworzą zamknięty łańcuch (kaskadę) elementów.



Wianek ze stokrotek - *daisy chain*

## SPI, JTAG, MIDI

## Struktura logiczna kaskadowa zamknięta, przykład SPI



**SCLK** — **S**erial **C**lock (output from master)

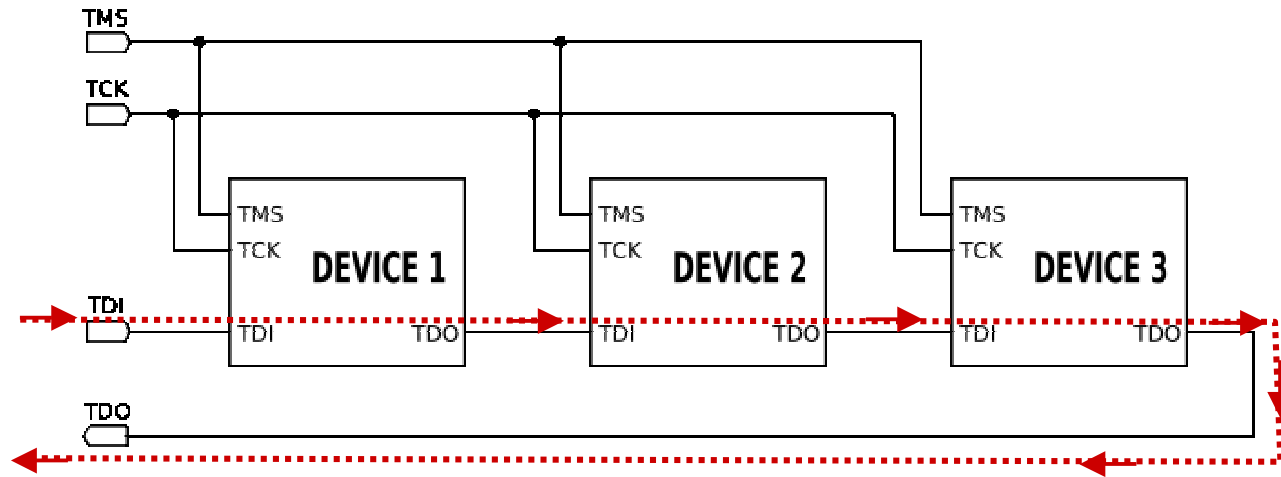
**MOSI/SIMO** — **M**aster **O**utput, **S**lave **I**nput (output from master)

**MISO/SOMI** — **M**aster **I**nput, **S**lave **O**utput (output from slave)

**SS** — **S**lave **S**elect (active low; output from master)

**SPI** (*Serial Peripheral Interface Bus*) w konfiguracji kaskady

## Struktura logiczna kaskadowa zamknięta, przykład JTAG

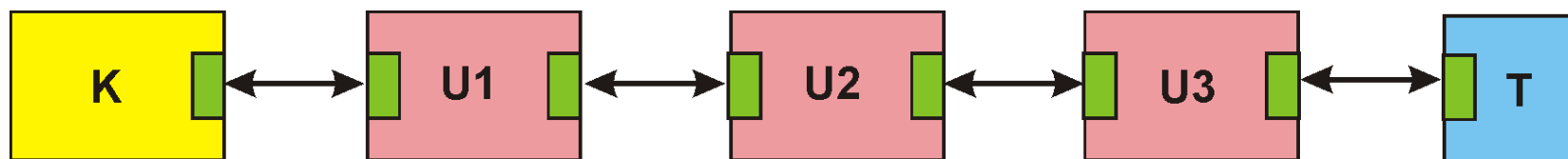


**TDI** (Test Data In)  
**TDO** (Test Data Out)  
**TCK** (Test Clock)  
**TMS** (Test Mode Select)  
**TRST** (Test Reset) optional.

**JTAG** (*Joint Test Action Group*) – konfiguracja *daisy chain*

## Struktura logiczna kaskadowa otwarta (*daisy chain - linear*)

Wszystkie urządzenia **U1**, **U2**, **U3**, ... oraz kontroler **K** tworzą łańcuch (kaskadę) elementów zakończony (opcjonalnie) terminatorem **T**



IEEE 1284 (rozszerzony Centronics) w trybie ECP/EPP

## Struktury połączeń fizycznych w systemach interfejsu

**Fizyczna struktura połączeń** opisuje „mapę połączeń” w systemie interfejsu - sposób okablowania, bez względu na logiczną organizację przepływu danych pomiędzy urządzeniami.

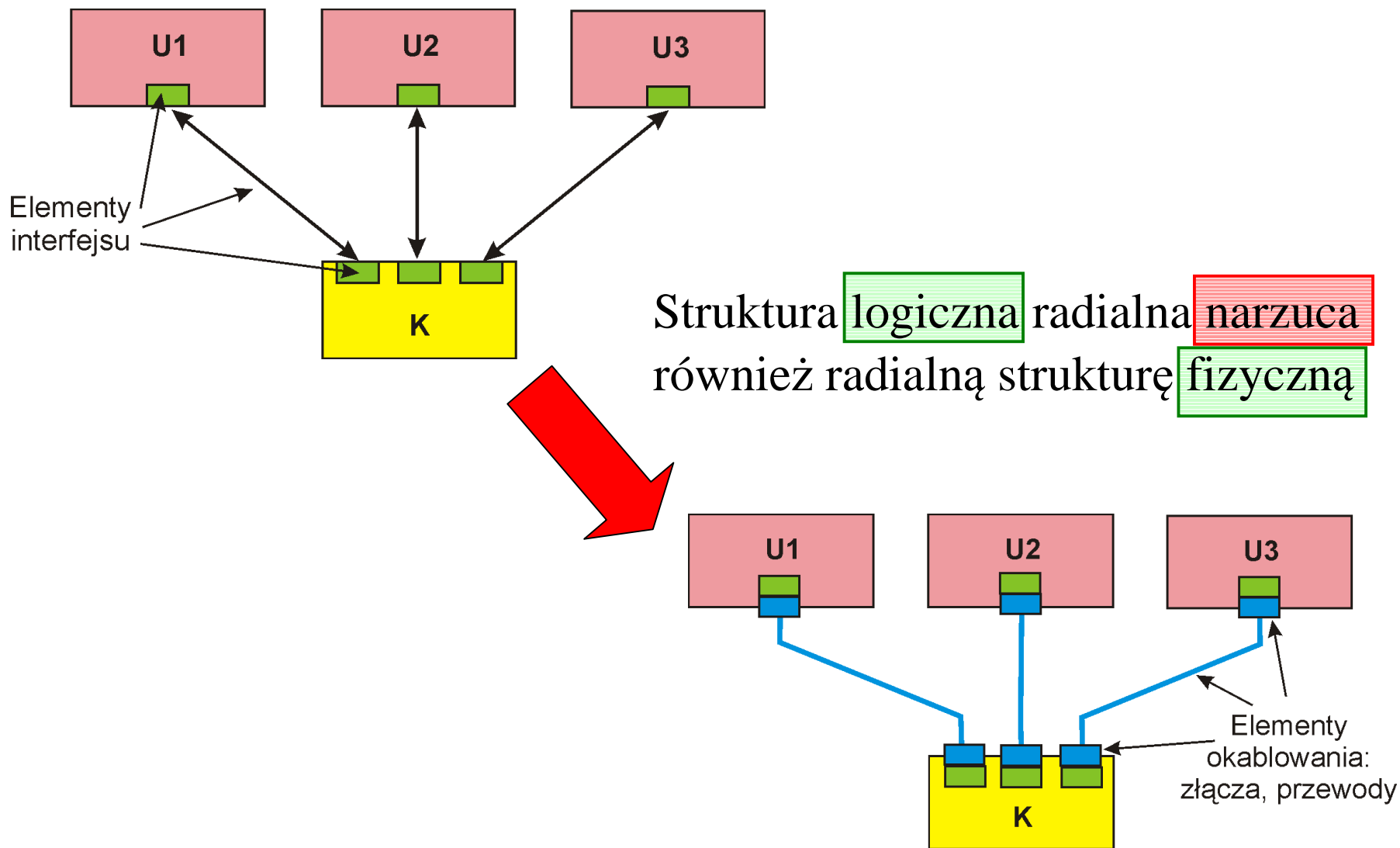
Logiczna struktura połączeń **w pewnym zakresie ogranicza** możliwe do zastosowania struktury fizyczne połączeń. Stosuje się **to samo nazewnictwo!**

**Przykłady:**

**Struktura logiczna radialna** – możliwa jest tylko również radialna struktura fizyczna (np. RS232),

**Struktura logiczna magistrala** – możliwa struktura fizyczna typu radialnego, drzewa, magistrali (IEEE 488).

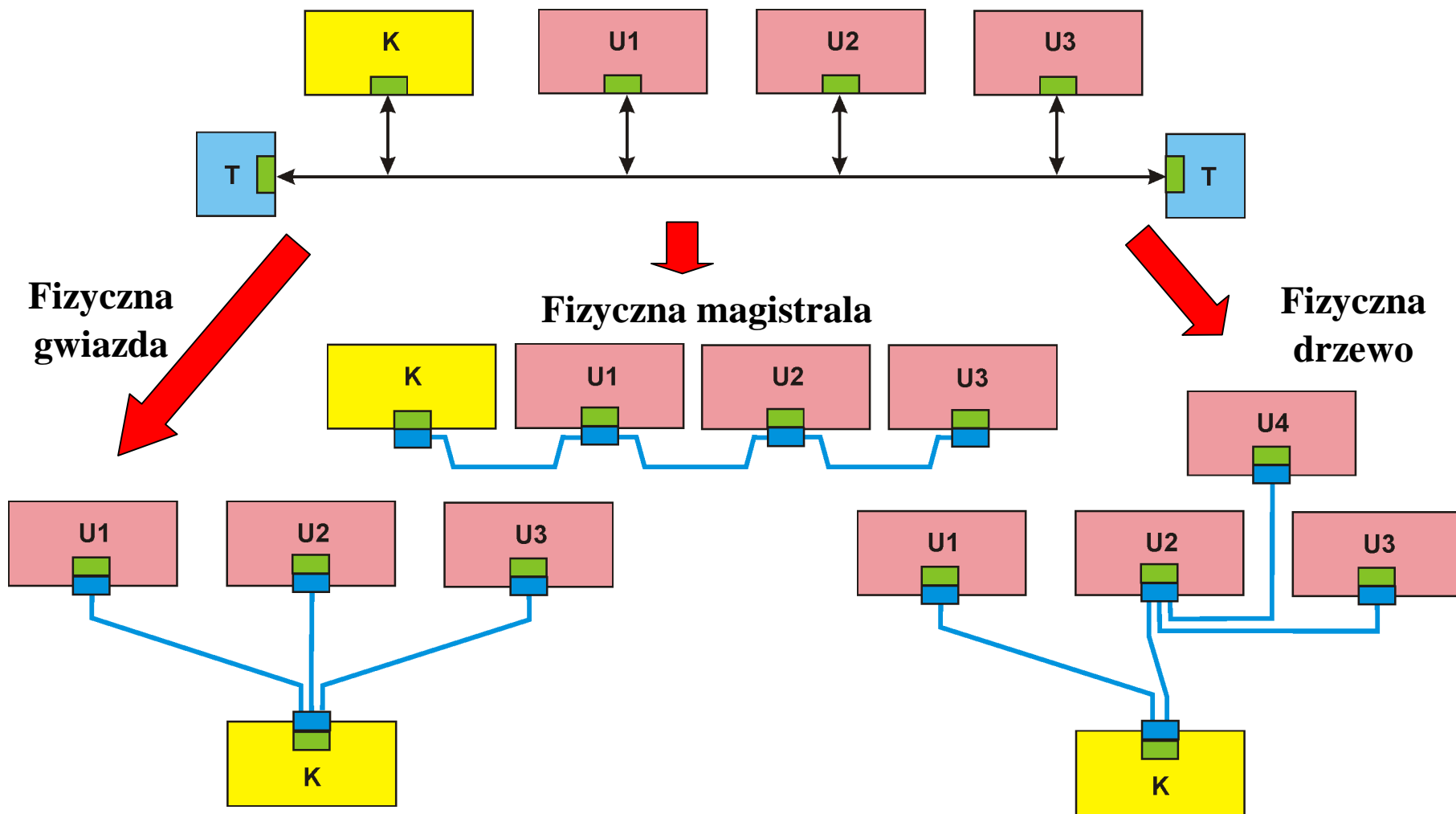
# Struktury połączeń fizycznych - przykład RS232





# Struktury połączeń fizycznych - przykład IEEE 488

Struktura logiczna magistrala umożliwia różne struktury fizyczne



## Sposoby transmisji porcji informacji w systemach interfejsu

Sposoby transmisji porcji informacji – bitów, bajtów i ramek danych:

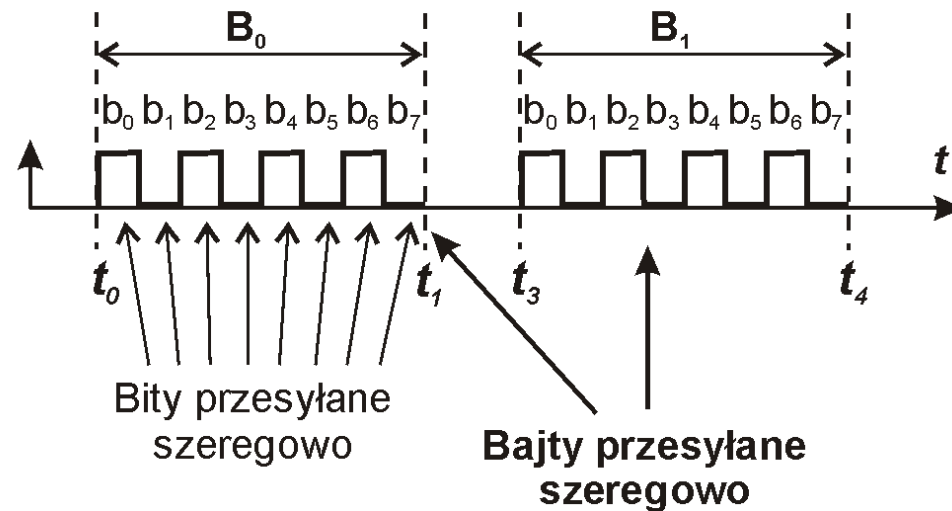
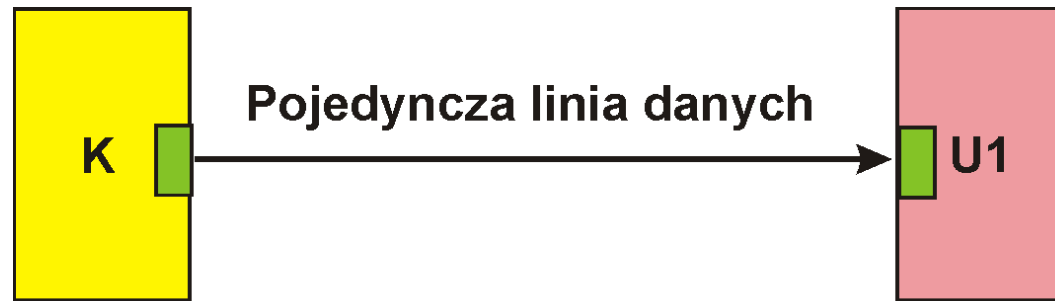
**1–Transmisja szeregowo:** bity, bajty i ramki szeregowo,

**2–Transmisja równoległa:** bity równoległe (jeden lub kilka bajtów jednocześnie),

**3–Transmisja szeregowo-równoległa:** bity równoległe, bajty i ramki szeregowo.

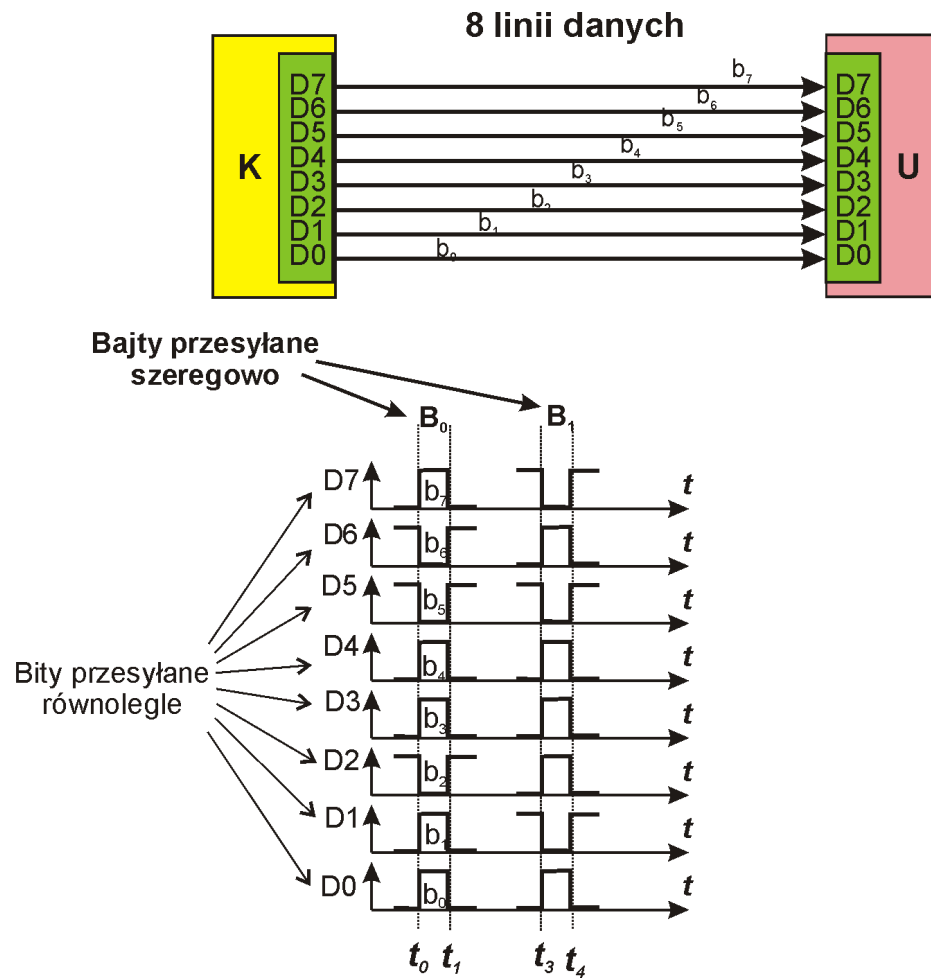
**W praktyce transmisja całkowicie równoległa (bity, bajty i ramki równoległe) nie jest możliwa do zrealizowania.**

# Transmisja szeregowo



**Transmisja szeregowo:** bity, bajty i ramki szeregowo (np.RS232),

# Transmisja równoległa (szeregowo-równoległa)



**Transmisja szeregowo-równoległa:** bity równoległe, bajty i ramki szeregowo (np. IEEE 488).

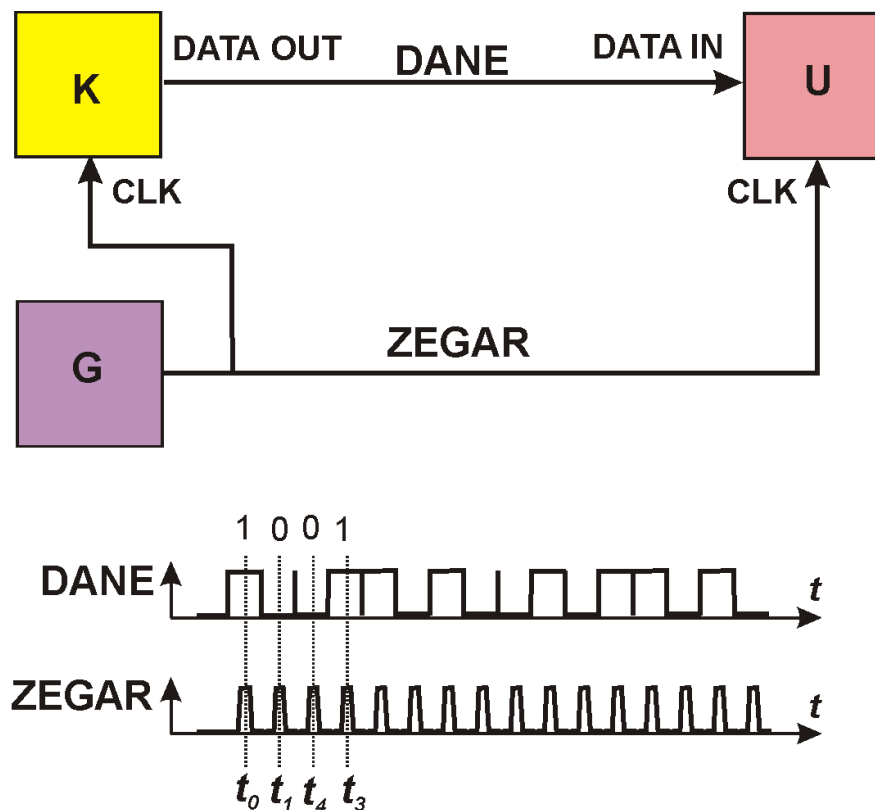
## Czasowa koordynacja transmisji w systemach interfejsu

Sposoby transmisji zapewniające **czasową koordynację** przesyłu danych pomiędzy nadajnikiem i odbiornikiem:

**1–Transmisja synchroniczna:** z transmisją sygnału zegarowego oddzielną linią lub jednocześnie wspólną linią z danymi,

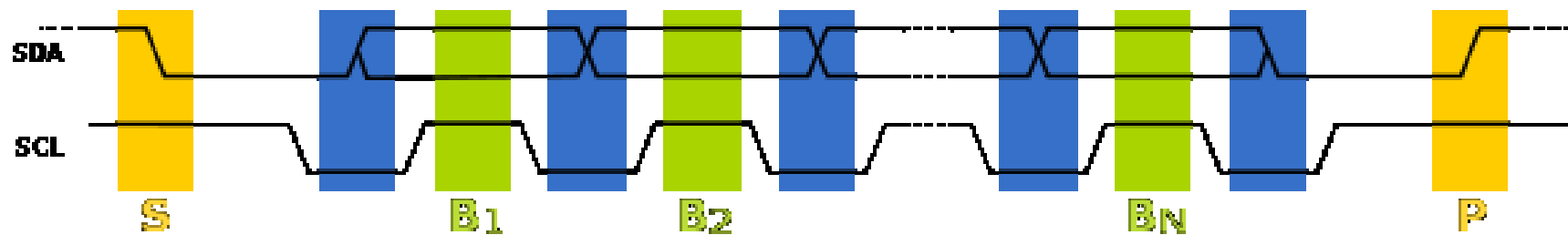
**2–Transmisja asynchroniczna:** z synchronizacją programową (start-stopową) lub sprzętową (*ang. handshaking*).

## Transmisja synchroniczna z oddzielną linią zegarową



**Transmisja synchroniczna:** dane DATA transmitowane synchronicznie z sygnałem zegarowym CLK przesyłanym oddzielną linią (aktywne jedno ze zboczy lub poziom sygnału).

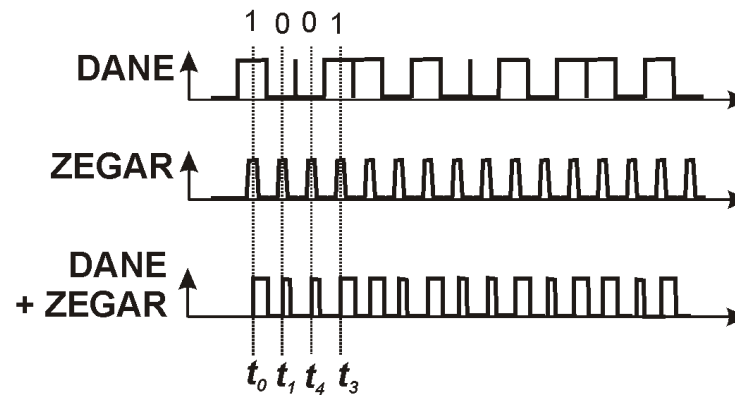
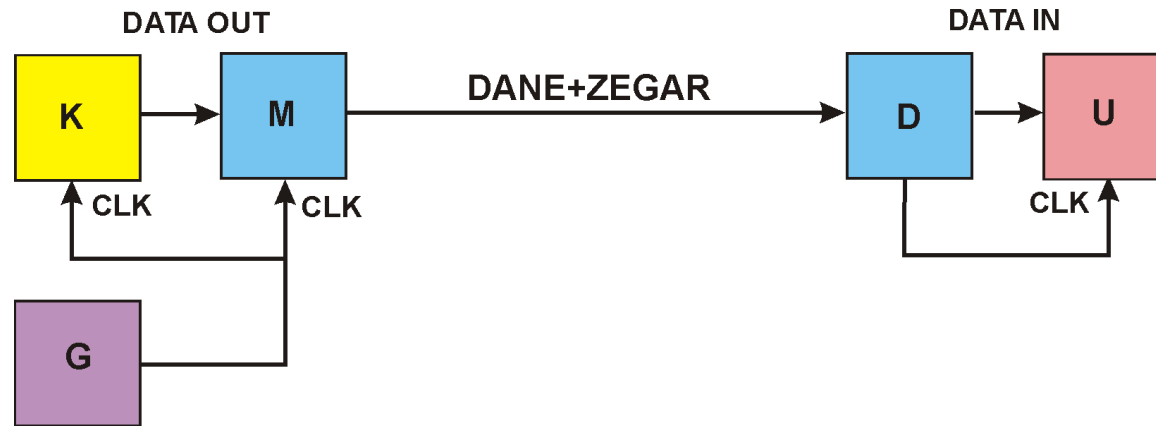
## Transmisja synchroniczna – przykład I2C



**Transmisja synchroniczna I2C:** dane transmitowane są linią **SDA** (*Serial DA*ta line) synchronicznie z sygnałem zegarowym przesyłanym oddzielną linią **SCL** (*Serial CL*ock line).

Kolejne bity  $b_1$ ,  $b_2$ , ...  $b_n$  są przesyłane w czasie trwania stanu wysokiego sygnału zegarowego. Dodatkowe kombinacje S i P rozpoczynają i kończą transmisję całego bajtu danych

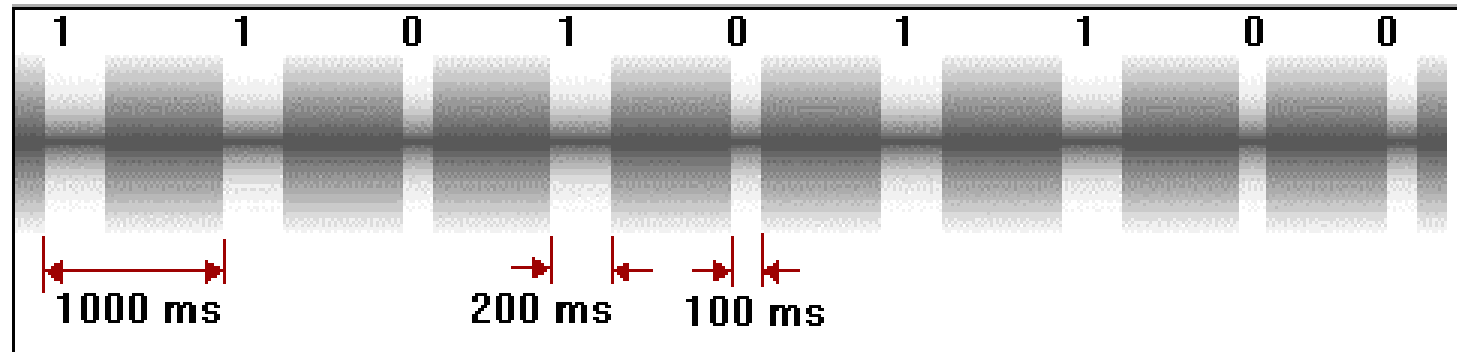
# Transmisja synchroniczna bez linii zegarowej



**Transmisja synchroniczna:** dane DATA transmitowane synchronicznie z sygnałem zegarowym CLK przesyłanym tą samą linią dzięki modulatorowi **M** i demodulatorowi **D**.

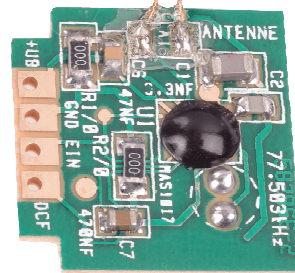
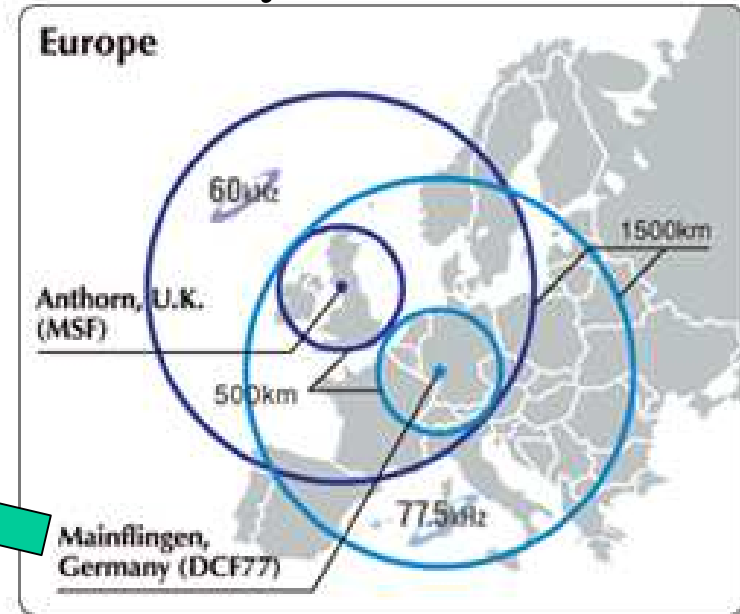


## Transmisja synchroniczna bez linii zegarowej-przykład DCF77



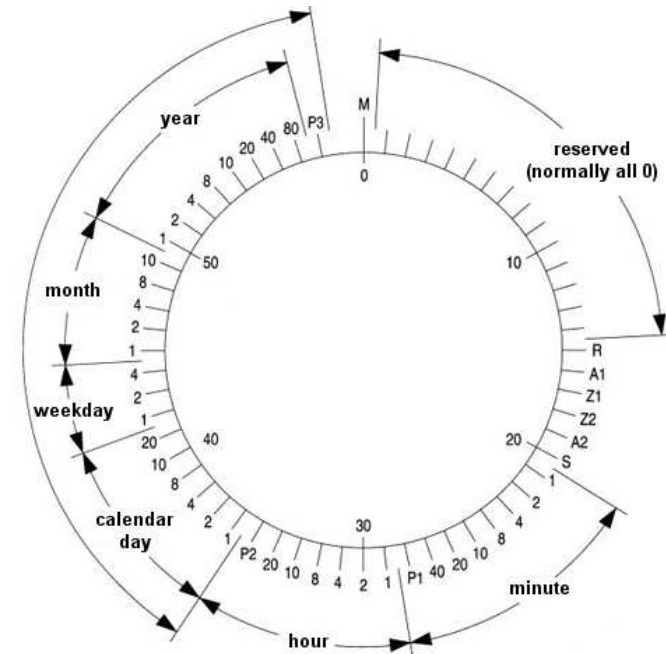
**Transmisja synchroniczna:** dane DATA transmitowane synchronicznie z sygnałem zegarowym CLK przesyłanym tym samym radiowym kanałem transmisji za pomocą modulacji amplitudy fali nośnej 77,5kHz.

# DCF77 – zasięg i kodowanie danych

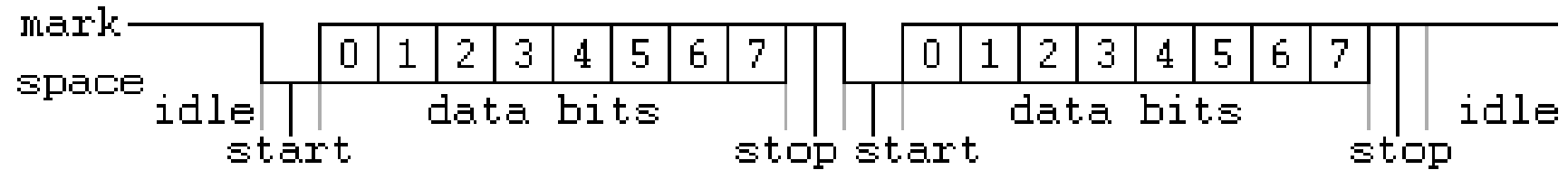


KSP, tydzień 4

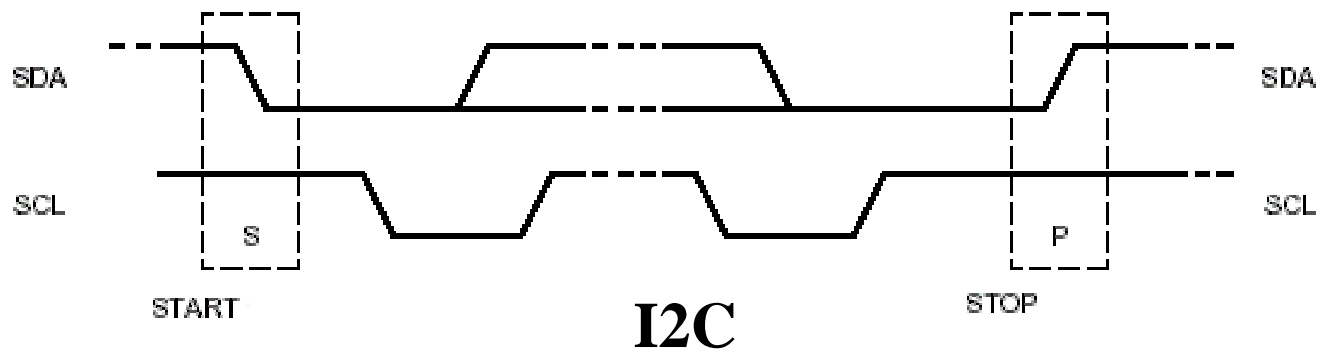
dr inż. Eligiusz Pawłowski



# Transmisja asynchroniczna z synchronizacją start-stopową



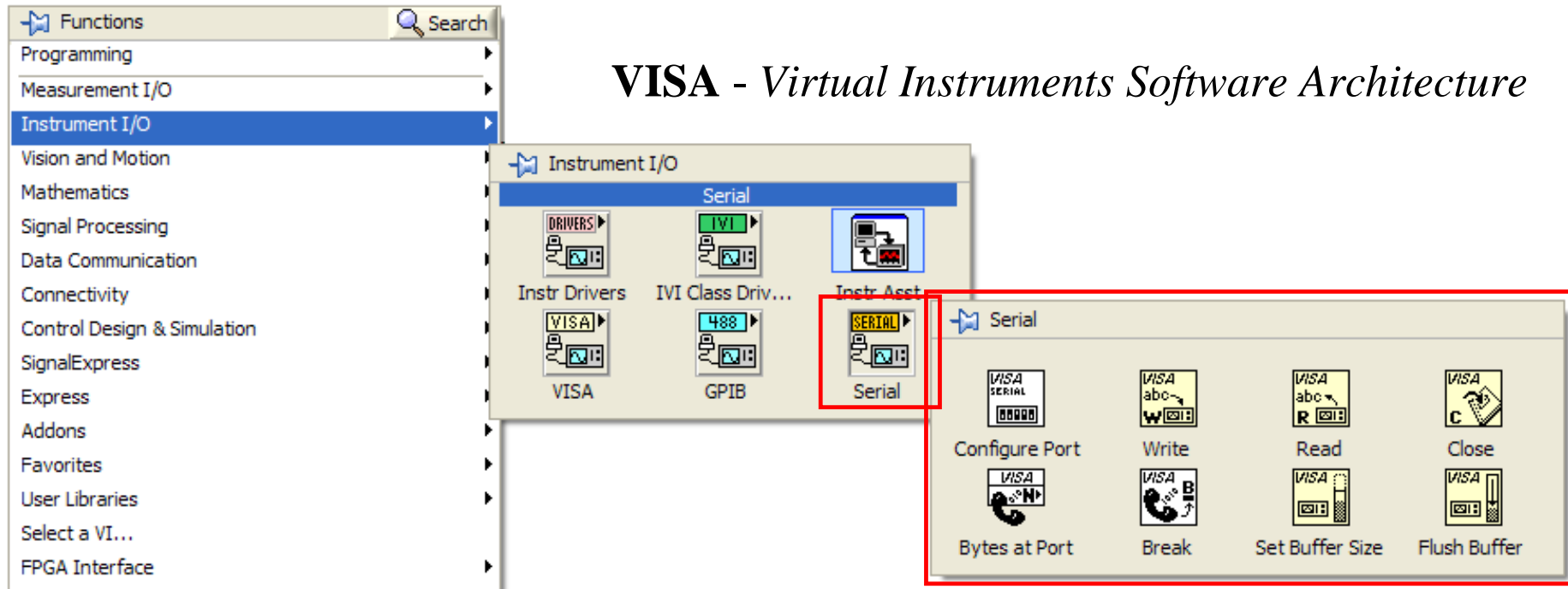
## RS232



**Transmisja asynchroniczna:** z synchronizacją start-stopową w interfejsie **RS232** i **I2C**

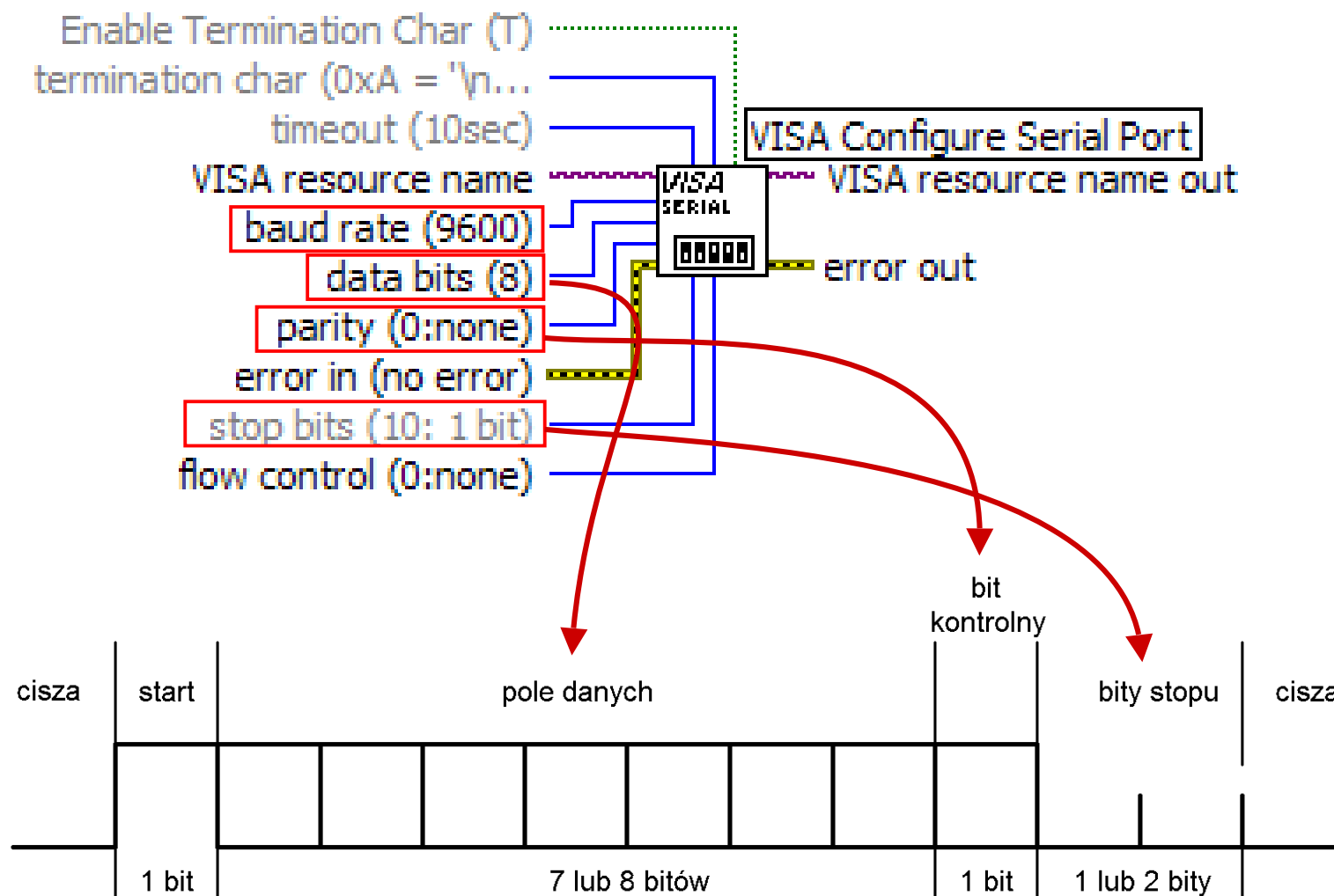
# Transmisja asynchroniczna RS232 w LabVIEW

*VISA - Virtual Instruments Software Architecture*



Funkcje VISA Instrument I/O do obsługi interfejsu RS-232C

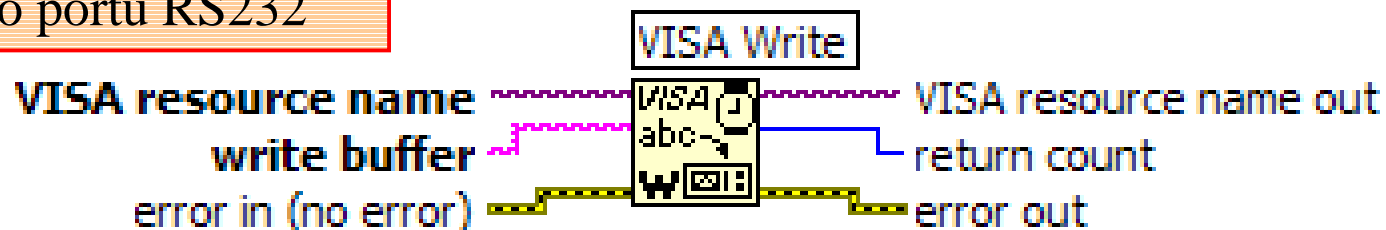
# Podstawowe funkcje VISA – konfiguracja portu RS232



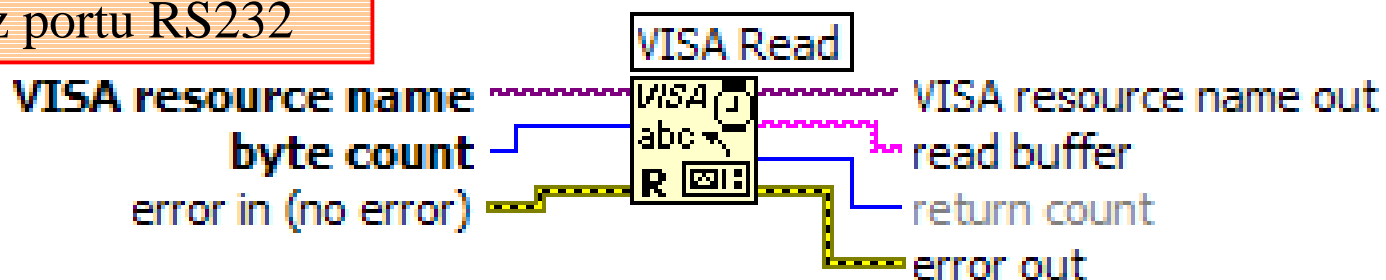
Struktura ramki RS232

# Zapis, odczyt i zamknięcie portu RS232

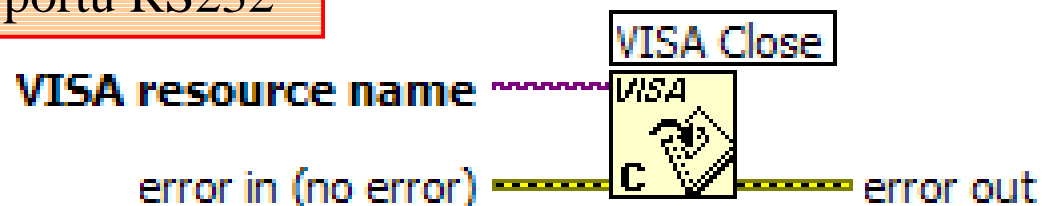
## Zapis do portu RS232



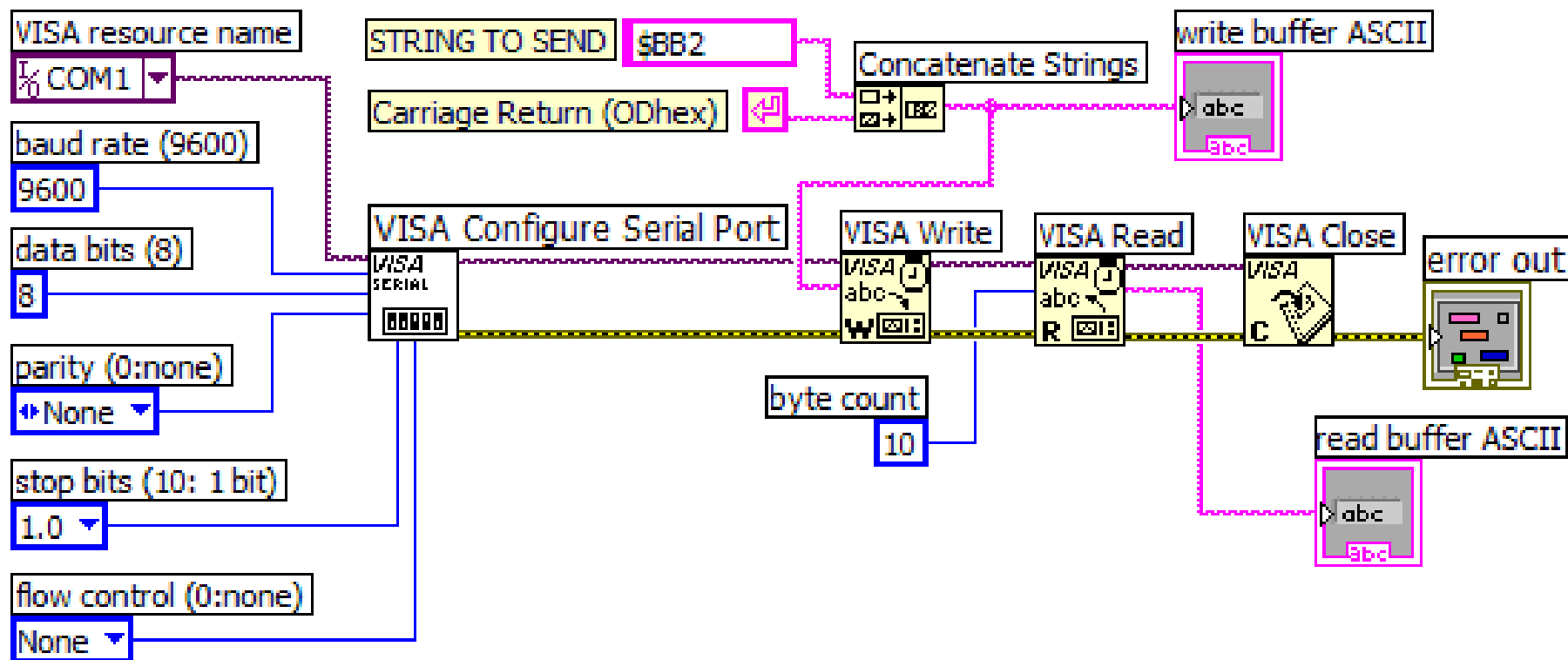
## Odczyt z portu RS232



## Zamknięcie portu RS232

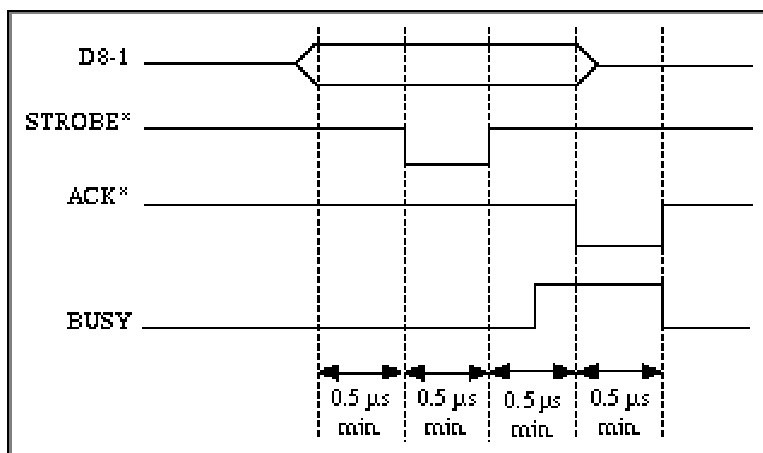
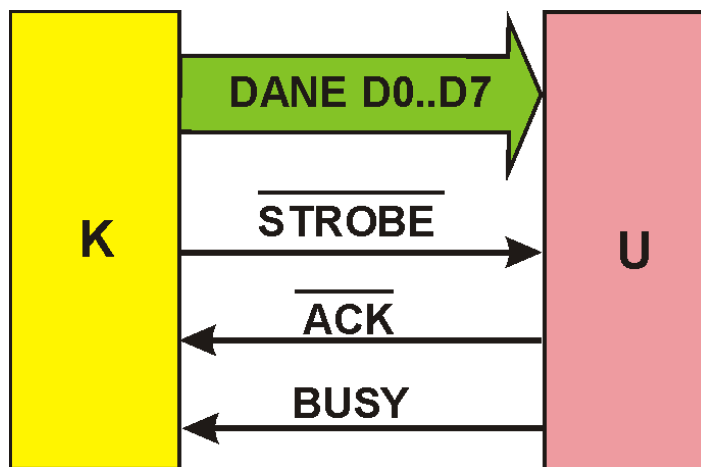


## Przykład pełnej obsługi portu RS232



Przykładowy Diagram programu obsługującego interfejs RS-232 do odczytu komendą *Configuration Status* konfiguracji modułu ADAM-4013 pod adresem BBhex (ćwiczenie nr 3 Laboratorium KSP)

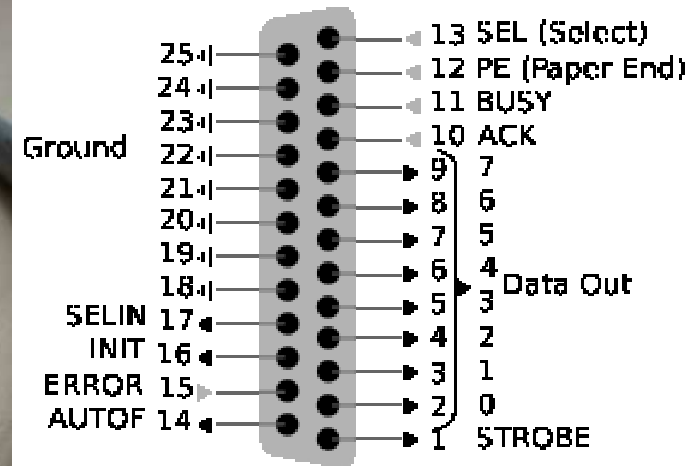
## Transmisja asynchroniczna z synchronizacją sprzętową



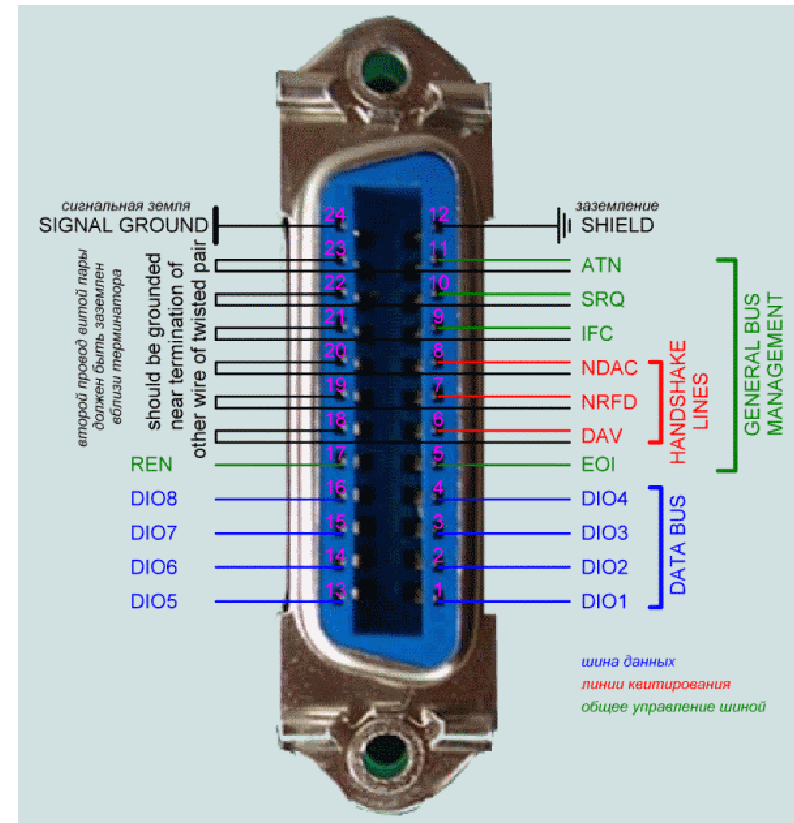
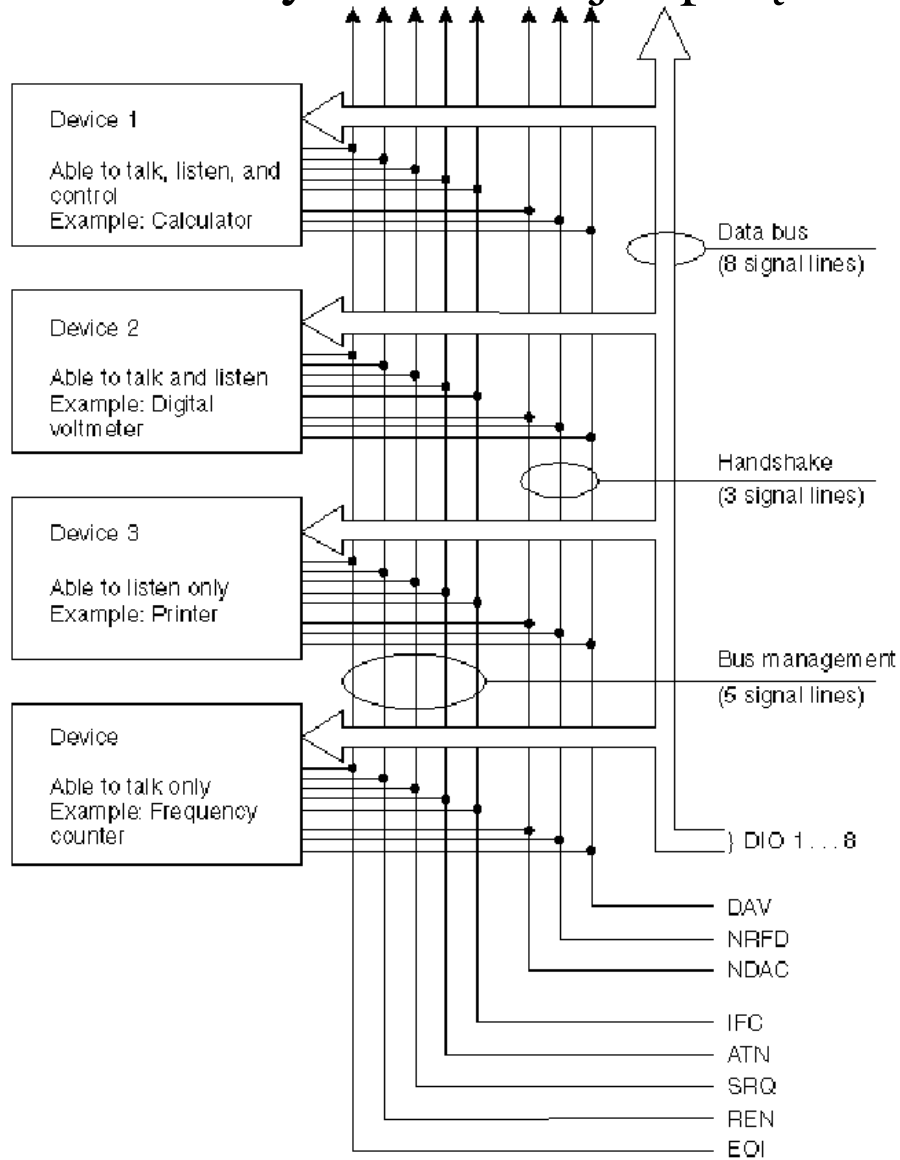
**Transmisja asynchroniczna:** z synchronizacją sprzętową (*ang. handshaking*) w interfejsie Centronics w trybie SPP.



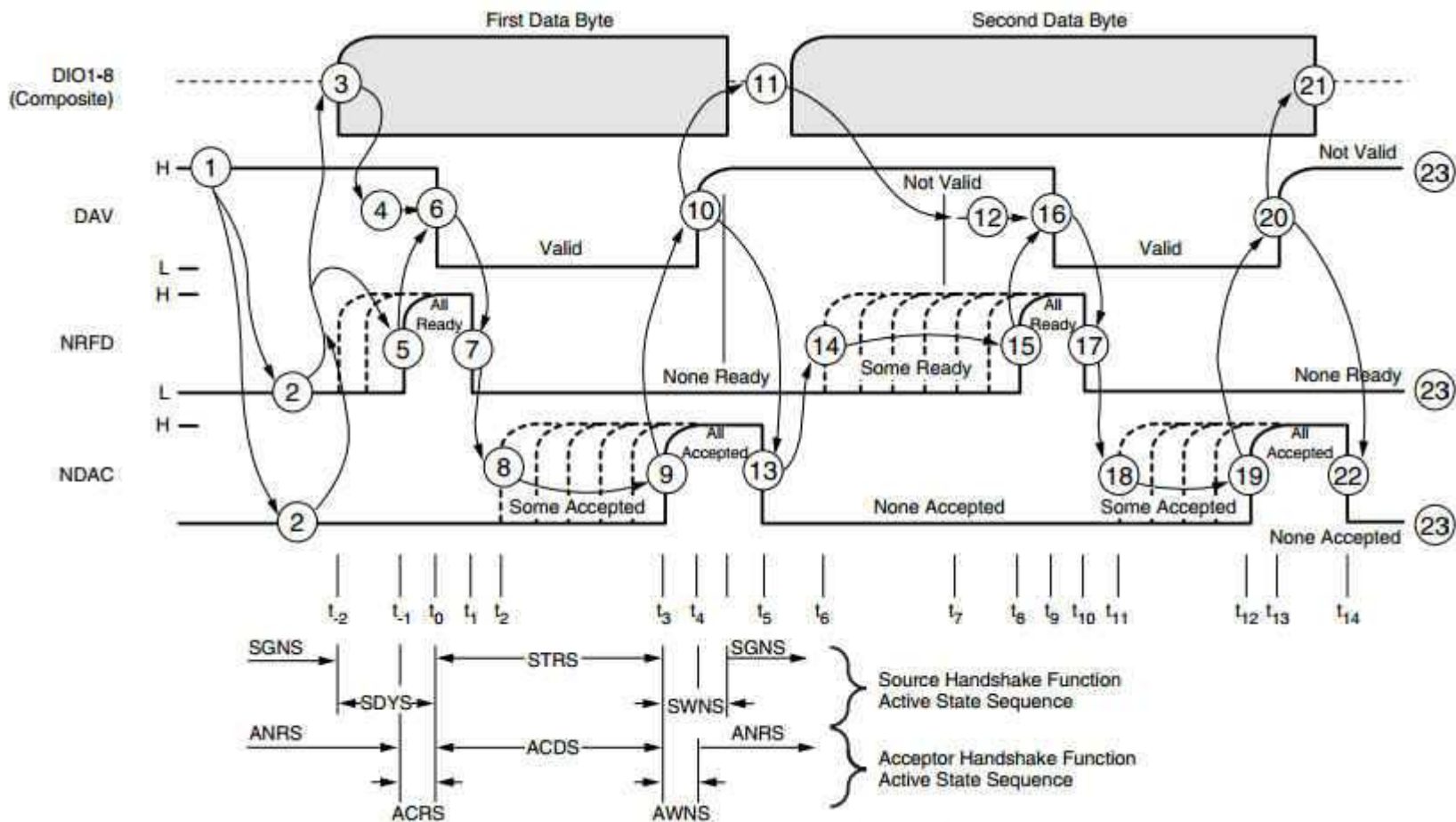
# Centronics – sygnały i okablowanie



# Synchronizacja sprzętowa IEEE 488 - sygnały

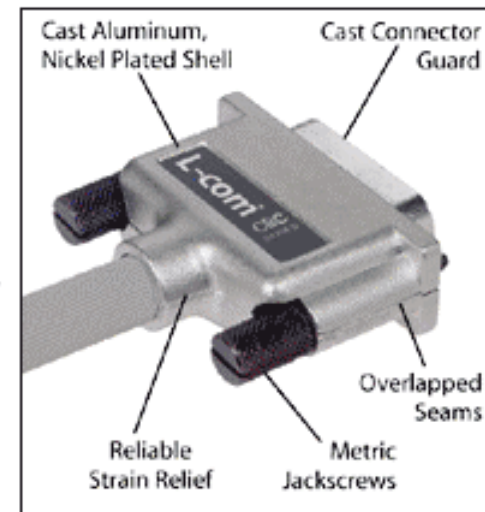
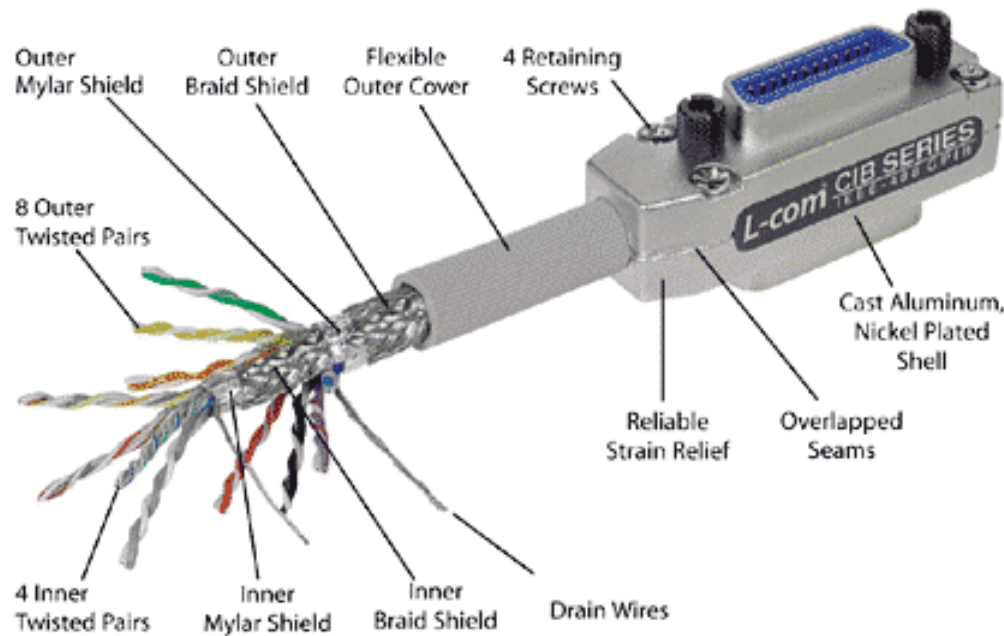
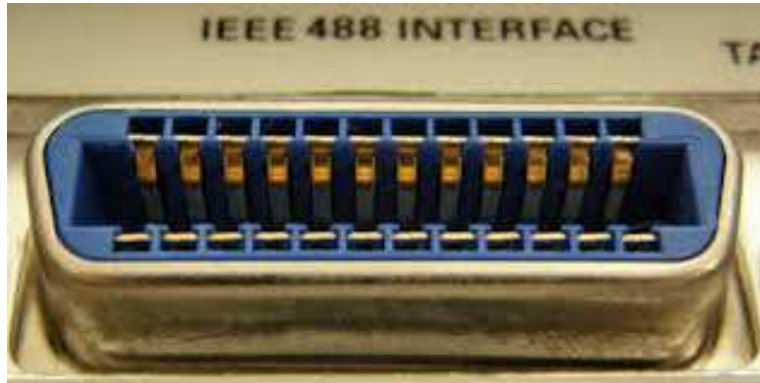


# Synchronizacja sprzętowa IEEE 488 - przebiegi



NOTE—(See Figure B.2 and List of Events)  $H \geq +2.0\text{ V}$ ;  $L \leq +0.8\text{ V}$ .

# IEEE 488 - okablowanie



## Adresowanie urządzeń w systemach interfejsu

Sposoby **adresowania urządzeń** w systemach interfejsu:

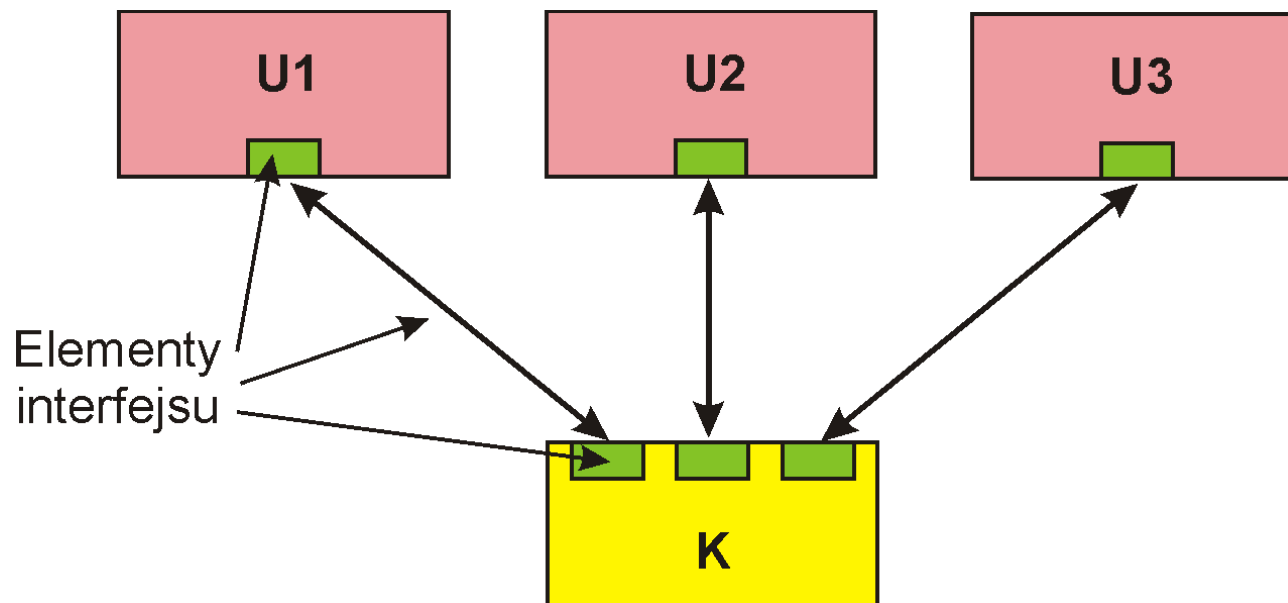
**1–Struktura logiczna gwiazdzista** – oddzielny interfejs dla każdego urządzenia,

**2–Adresowanie sygnałami wyboru urządzenia** – oddzielny sygnał wyboru CS (*ang. Chip Select*) dla każdego urządzenia,

**3–Adresowanie magistralą adresową** – przesyłanie adresu równoległe magistralą adresową,

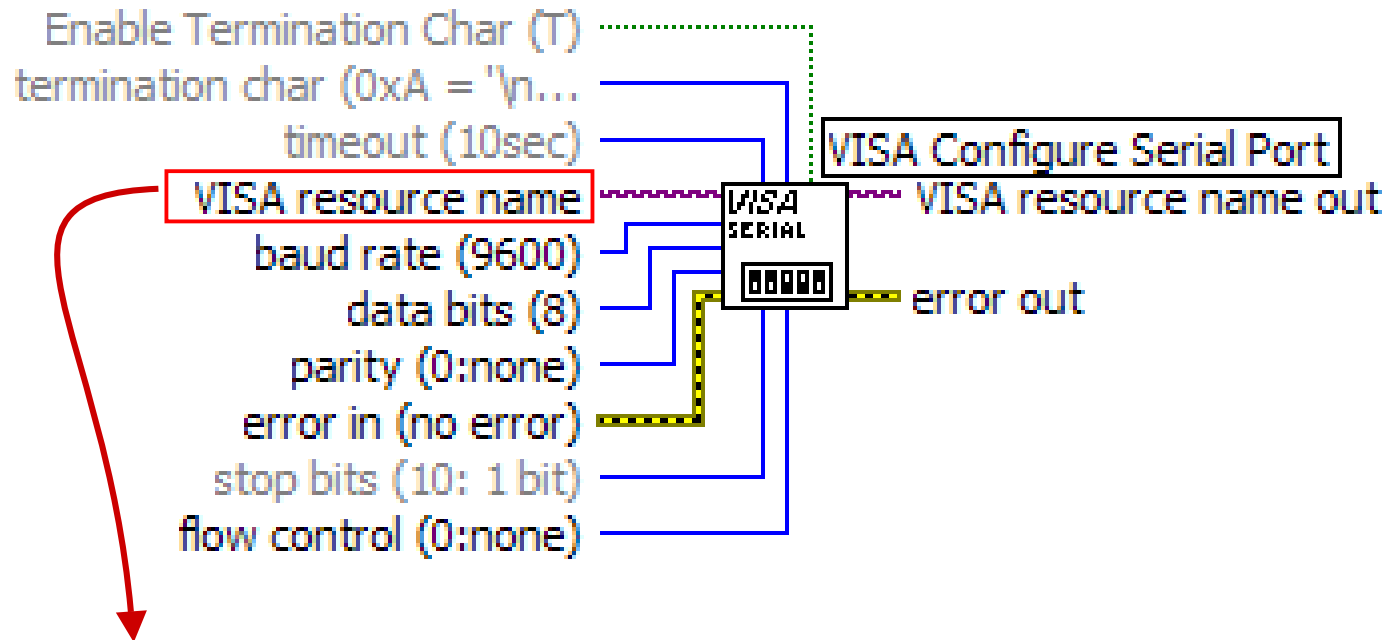
**4–Adresowanie polem adresowym w ramce danych** – przesyłanie adresu szeregowo w polu adresowym ramki danych.

## Adresowanie urządzeń w strukturze gwiazdziej



**Struktura logiczna gwiazdzista** – adresowanie urządzeń jest **zbędne**, każde urządzenie dołączone jest do kontrolera oddzielnym interfejsem, nie ma możliwości wystąpienia konfliktu przy przesyłaniu danych, adresowanie oznacza wybór portu.

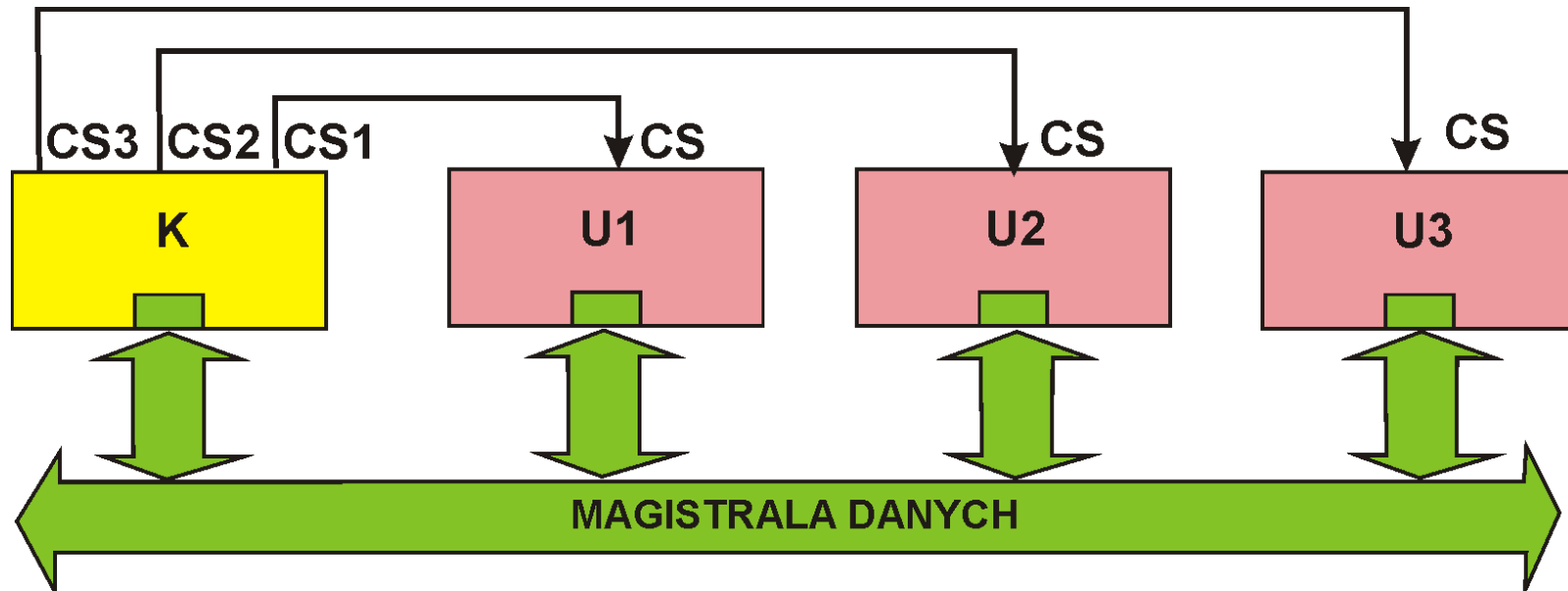
## Adresowanie w interfejsie RS232 poprzez wybór portu



**Struktura logiczna gwiazdzista, interfejs RS232**  
adresowanie oznacza wybór portu do którego  
dołączone jest adresowane urządzenie

### Struktura ramki RS232

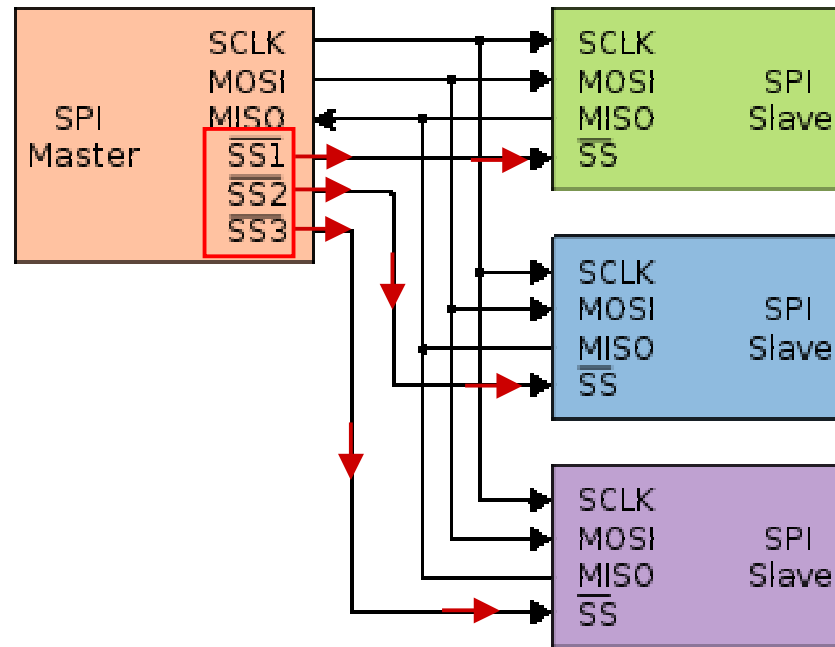
## Adresowanie urządzeń sygnałami wyboru CS



**Adresowanie sygnałami wyboru urządzenia** – oddzielny sygnał wyboru CS (*ang. Chip Select*) dla każdego urządzenia, dane przesyłane magistralą danych.

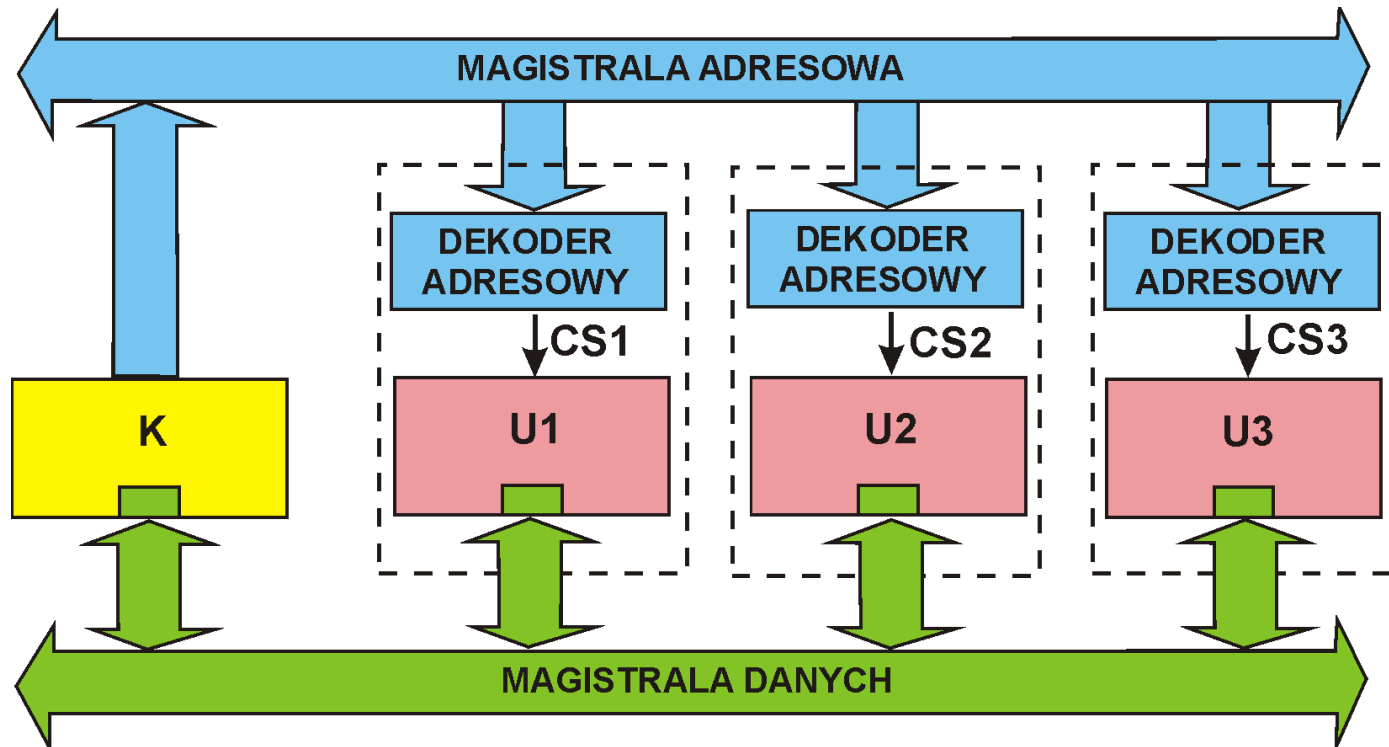


## Adresowanie urządzeń sygnałami wyboru CS - przykład SPI



**Adresowanie sygnałami wyboru urządzenia** – oddzielny sygnał wyboru CS dla każdego urządzenia w systemie SPI (*Serial Peripheral Interface Bus*) w konfiguracji magistrali, sygnały wyboru oznaczone SS1, SS2, SS3 (*Slave Select*), aktywny stan niski.

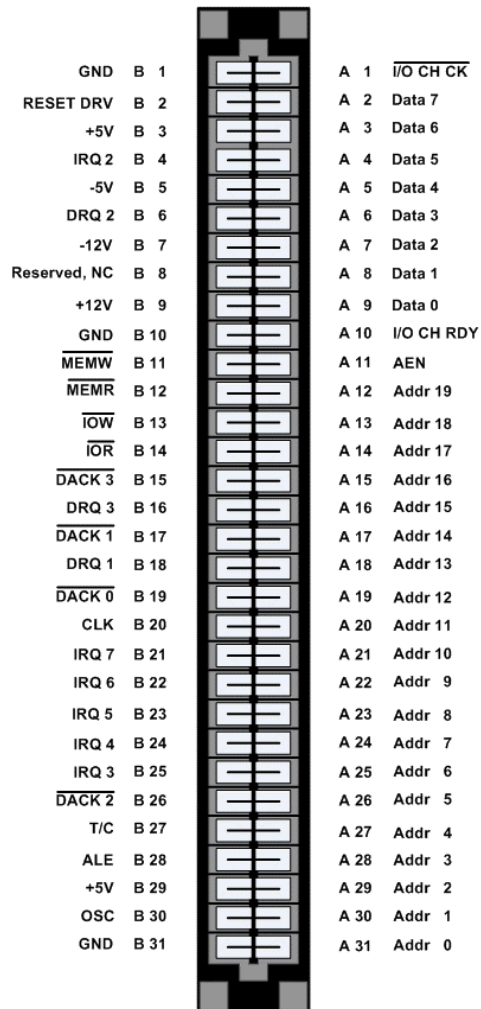
## Adresowanie urządzeń magistralą adresową



**Adresowanie magistralą adresową** – adres przesyłany jest równolegle magistralą adresową, każde urządzenie posiada własny dekodery adresowy dekodujący jego adres i generujący sygnał wyboru urządzenia **CS**.

# Adresowanie urządzeń magistralą adresową – ISA PC/XT

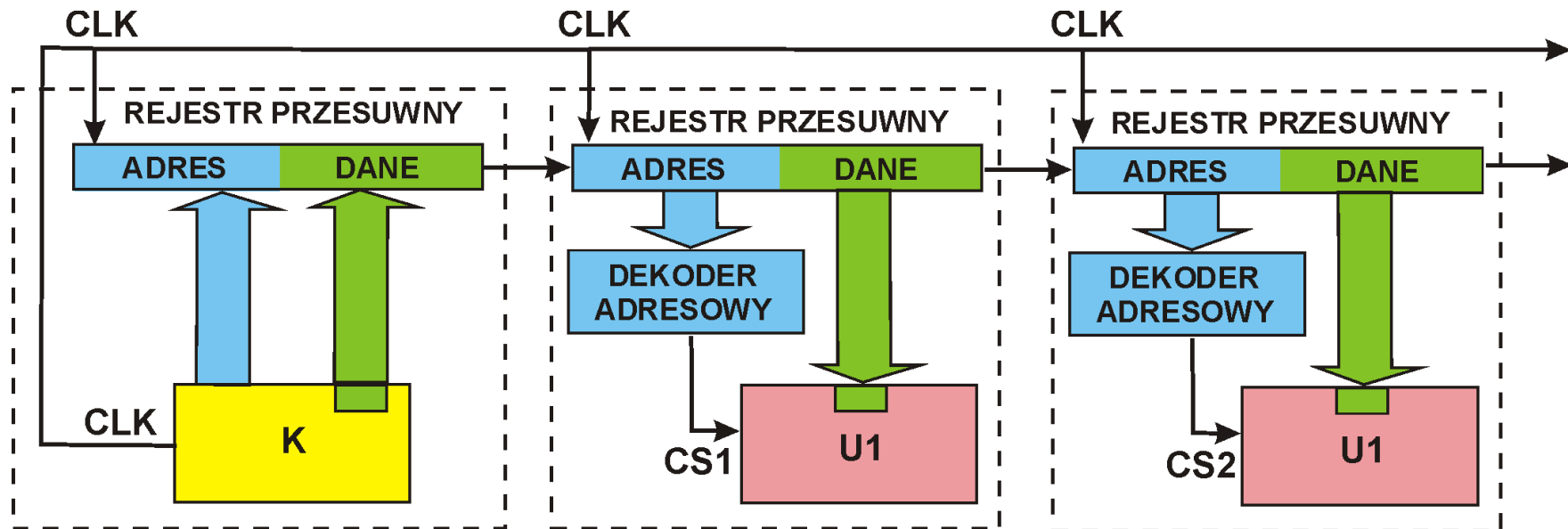
8 Bit XT Bus – top view



## Adresowanie magistralą adresową

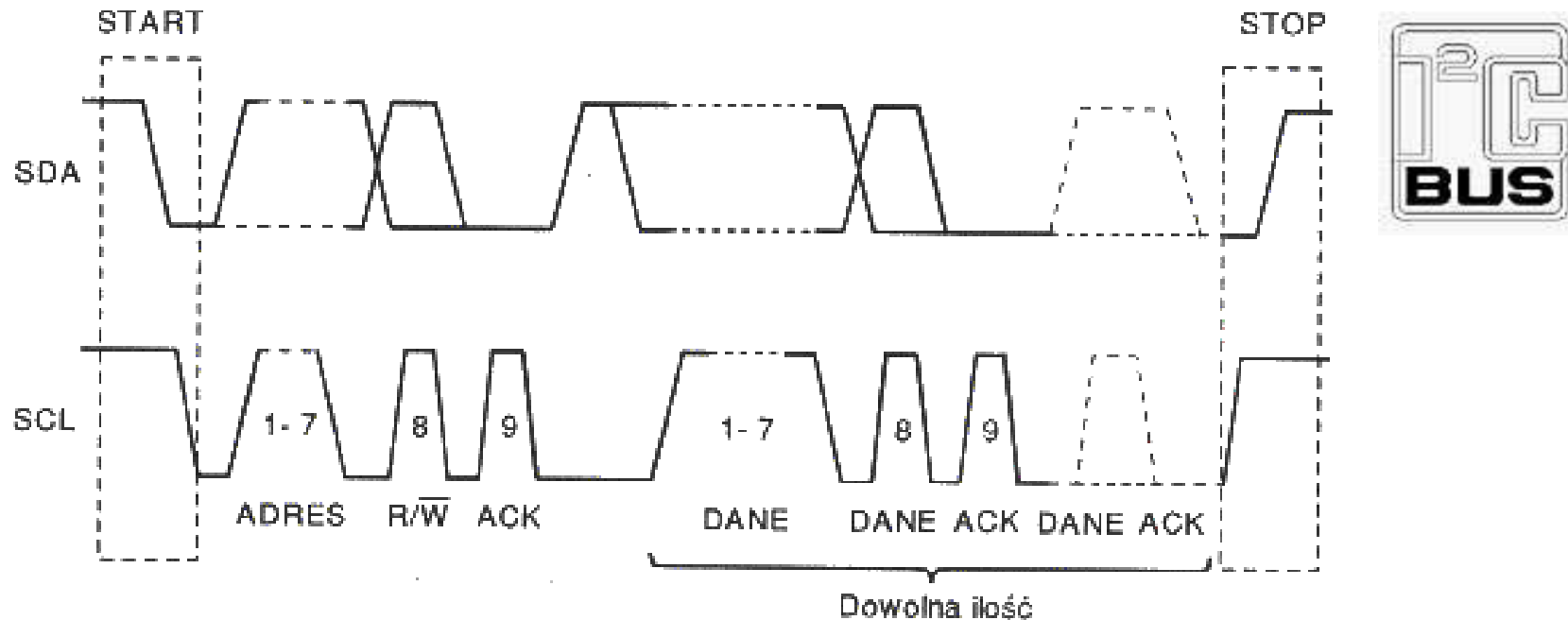
– przykład 8-bitowej magistrali ISA w komputerach IBM PC/XT - przesyłanie adresu równoległe magistralą adresową: 20 linii adresowych ADDR0 .. ADDR19 na stykach A31 .. A12 złącza krawędziowego 2x31 styków.

## Adresowanie urządzeń polem adresowym w ramce danych



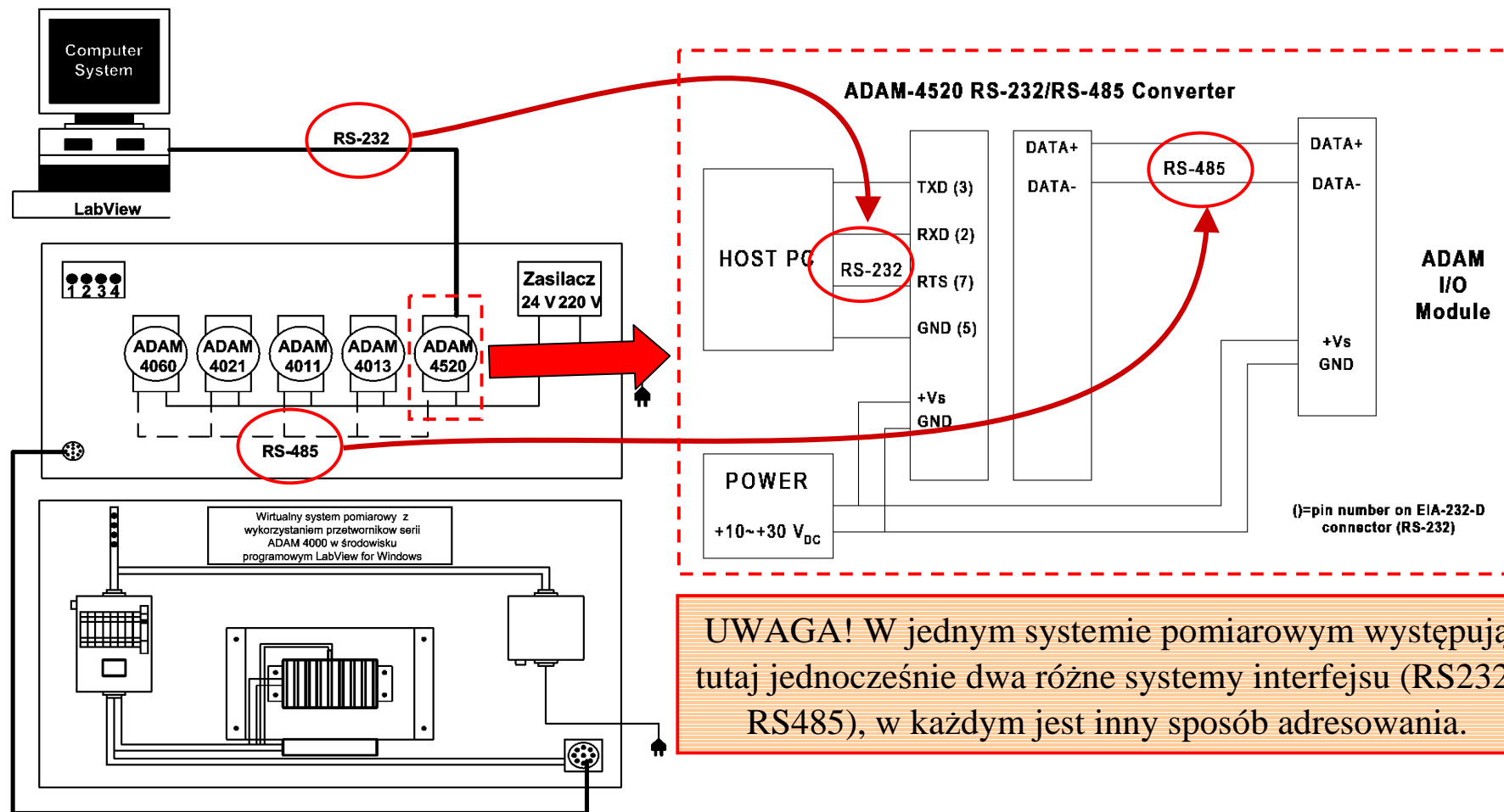
**Adresowanie polem adresowym w ramce danych** – przesyłanie adresu szeregowo w polu adresowym ramki danych.

## Adresowanie urządzeń polem adresowym w ramce danych I2C



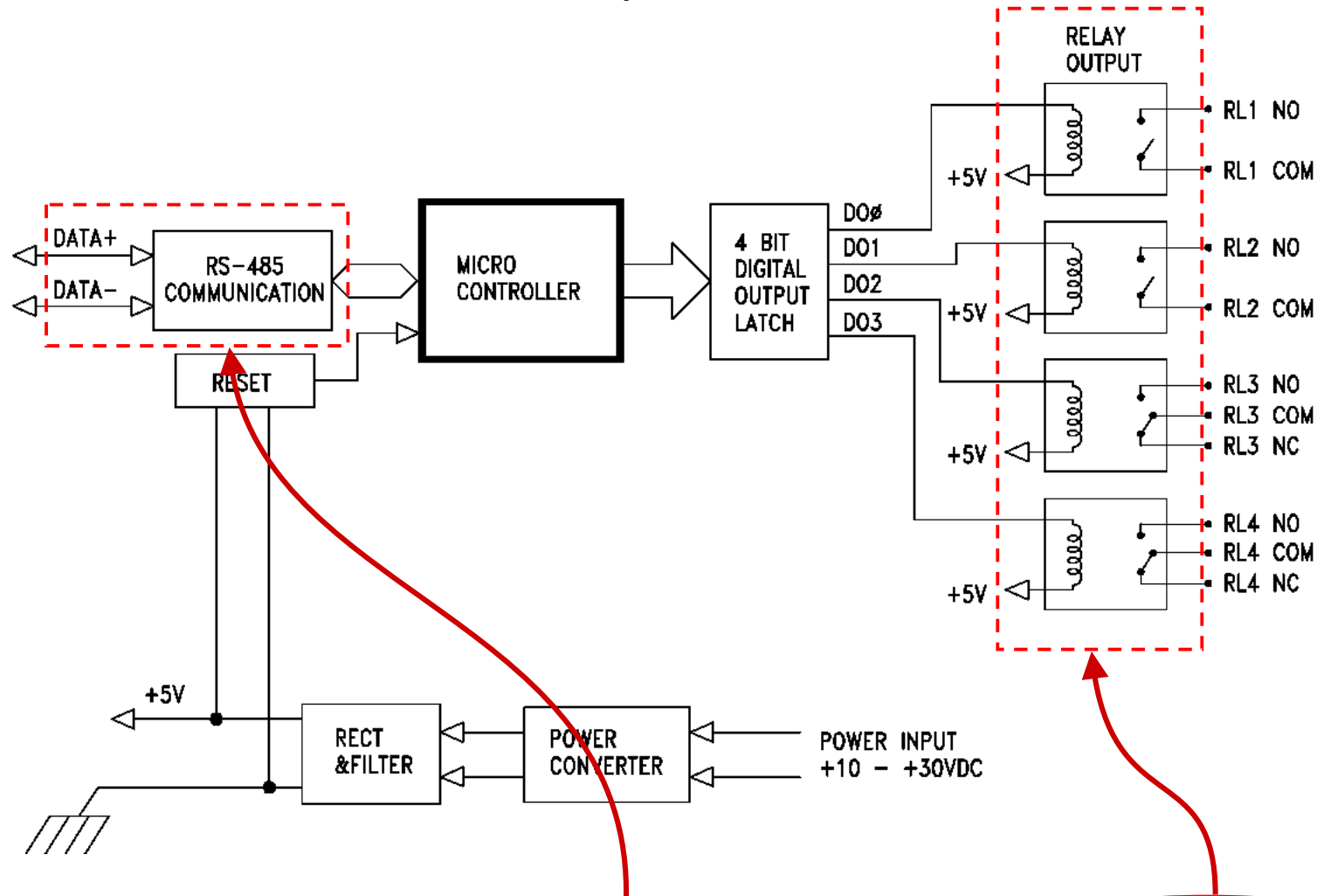
**Adresowanie polem adresowym w ramce danych** – przykład transmisji szeregowej w interfejsie I2C. Adres i dane transmitowane są linią **SDA** (*Serial DATA line*) synchronicznie z sygnałem zegarowym na linii **SCL** (*Serial CLOCK line*).

# Adresowanie polem adresowym w ramce danych RS485



Stanowisko dydaktyczne z modułami ADAM 4000 z interfejsem RS485 w konfiguracji **magistrali** (ćwiczenie nr 3 Laboratorium KSP)

# Schemat blokowy modułu ADAM 4060



Moduł ADAM 4060 z interfejsem RS485 do sterowania czterema przekaźnikami

# Komenda adresująca i programująca moduł ADAM4060

**#AABB(data)(CR)**

# - (hash) jest znakiem rozdzielającym (delimiter character) o kodzie ASCII 23hex.

**AA - jest polem adresowym zawierającym dwuznakowy adres modułu w kodzie szesnastkowym w zakresie 00-FF. Wykorzystywany w ćwiczeniu moduł ADAM-4060 posiada adres w zapisie szesnastkowym 64hex.**

**BB** - jest polem konfiguracyjnym sposób sterownia przekaźnikami. Jeśli BB=00 to możliwe jest sterowanie jednocześnie wszystkimi przekaźnikami, jeśli chcemy sterować pojedynczym przekaźnikiem, to pierwszy znak jest zawsze równy 1, a drugi jest numerem przekaźnika do sterowania (od 0 do 7).

**(data)** - jest dwuznakową liczbą szesnastkową reprezentującą cyfrowe dane wyjściowe:

-jeśli sterujemy pojedynczym przekaźnikiem, to pierwszy znak jest zawsze równy 0, a drugi jest równy 0 (wyłącza przekaźnik) lub 1 (załącza przekaźnik),

-jeśli sterujemy wszystkimi przekaźnikami jednocześnie, to dwuznakowa liczba szesnastkowa jest interpretowana w zapisie binarnym w ten sposób, że każdy bit steruje niezależnie jednym przekaźnikiem, szczegóły są widoczne na rys. 13.

**(CR)** - (Carriage return - powrót karetki) jest znakiem sterującym ASCII 0Dhex oznaczającym koniec komendy.

## Command Set

4050, 4060, 4055, 4068

### #AABB

**Name** Digital Data Out

**Description** The command either sets a single digital output channel or sets all digital output channels simultaneously.

**Syntax** #AABB(data)(cr)

# is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of the digital I/O module you want to set its output value.

BB is used to indicate whether all channels will be set or a single channel will be set. In the last case BB also indicates which channel. Writing to all channels (write a byte): both characters should be equal to zero (BB=00). Writing to a single channel (write a bit): First character is 1, second character indicates channel number which can range from 0 to 7.

(data) is the hexadecimal representation of the digital output value(s).

**When writing to a single channel (bit)** the first character is always 0. The value of the second character is either 0 or 1.

**When writing to all channels (byte)**, both characters are significant (range 00h-FFh). The digital equivalent of these two hexadecimal characters represent the channels values.

The amount of channels on the ADAM-4050, ADAM-4055, ADAM-4060 and ADAM-4068 differs. The value 7A would mean the following for the 8 channels on the ADAM-4050, ADAM-4055 and ADAM-4068:

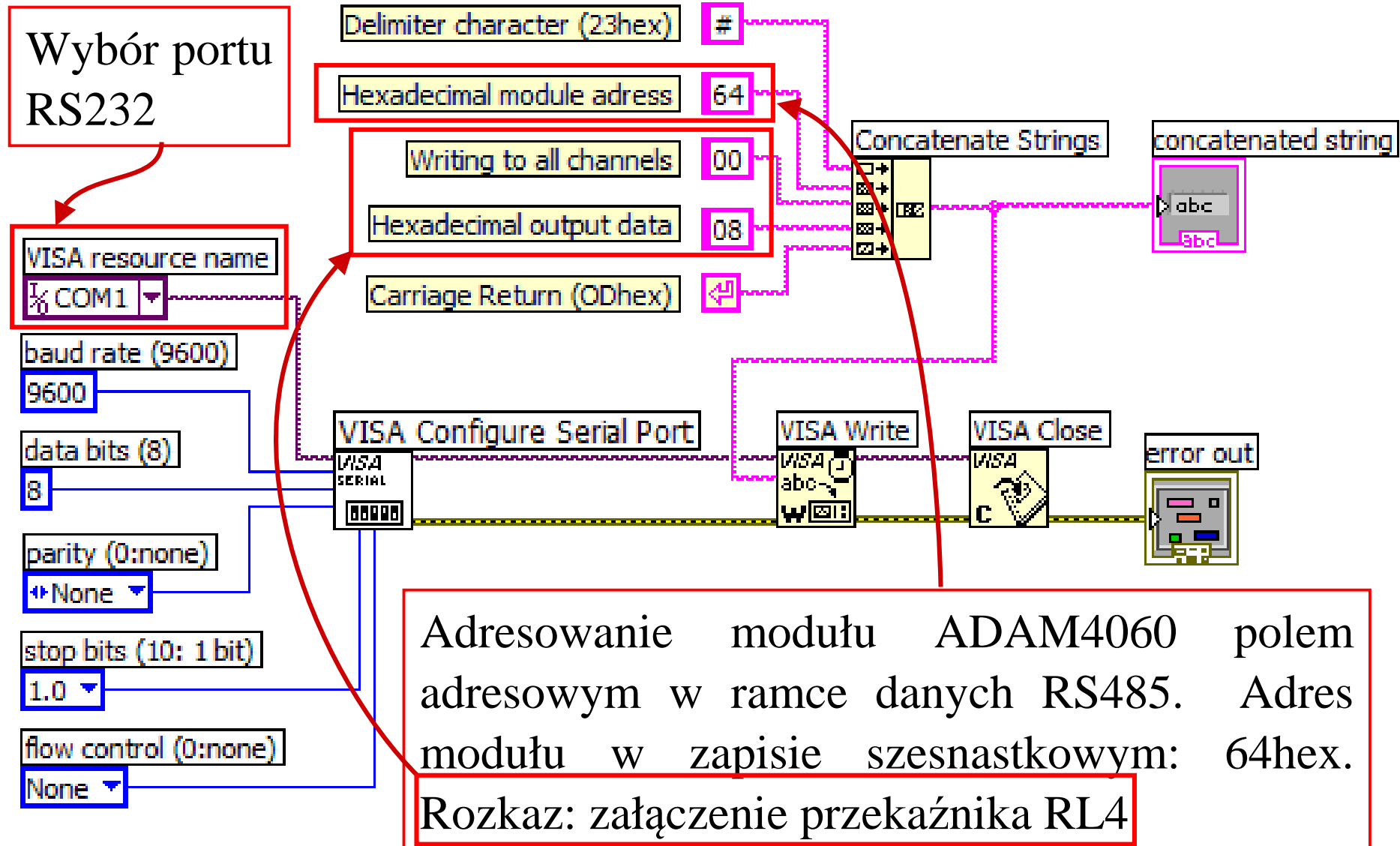
digital value:	0	1	1	1	1	0	1	0
ADAM-4050/4055/4068 channel no.	7	6	5	4	3	2	1	0

Since the ADAM-4060 has only four output channels all the meaning full values lie between 00h and 0Fh. The value 0Ah would mean the following for the ADAM-4060:

digital value:	0	0	0	0	1	0	1	0
ADAM-4060 channel no.	-	-	-	-	3	2	1	0



## Przykład obsługi modułu ADAM4060 w LabVIEW

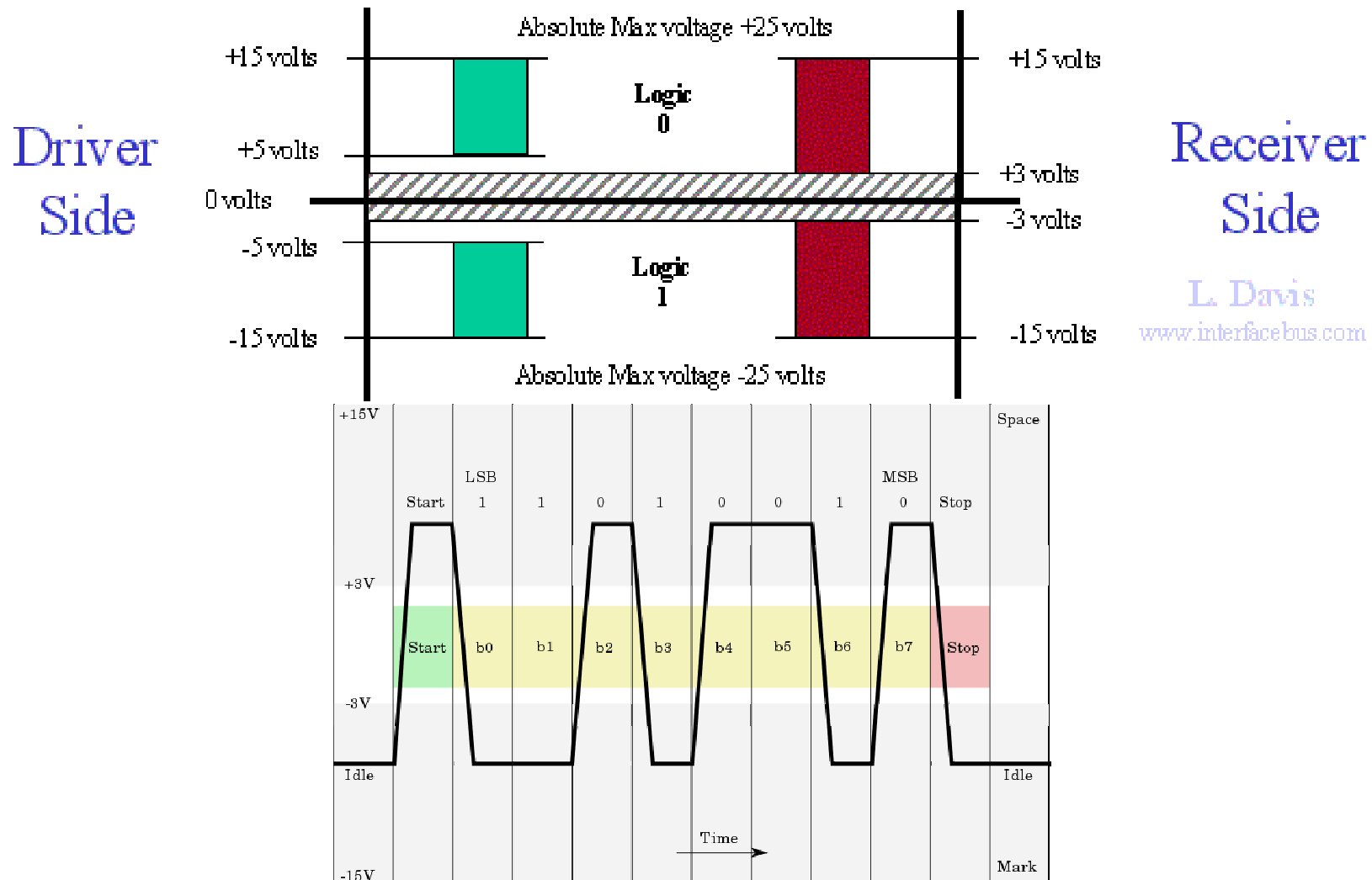


## Parametry elektryczne i mechaniczne systemów interfejsu

Do podstawowych parametrów **elektrycznych i mechanicznych** systemu interfejsu można zaliczyć:

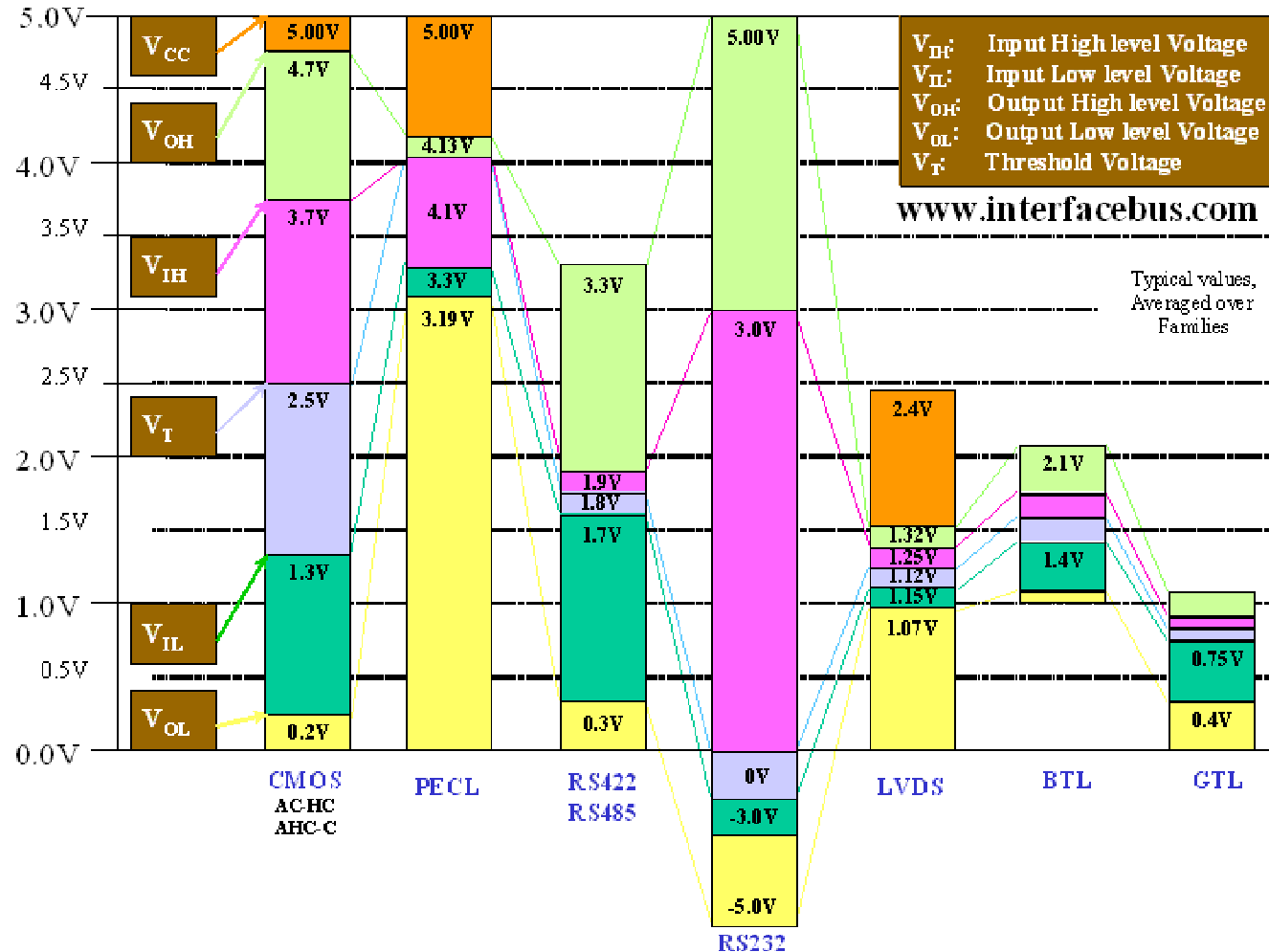
- 1–Rodzaj sygnału:** prądowy lub napięciowy,
- 2–Poziomy napięcie (prądów)** dla stanów logicznych 0 i 1,
- 3–Konfiguracja linii względem ziemi:** symetryczna lub niesymetryczna,
- 4–System złączy:** szufladowe, krawędziowe, koncentryczne,
- 5–Okablowanie:** skrętka, przewody koncentryczne, ekranowanie

# Poziomy napięć stanów logicznych standardowego RS232

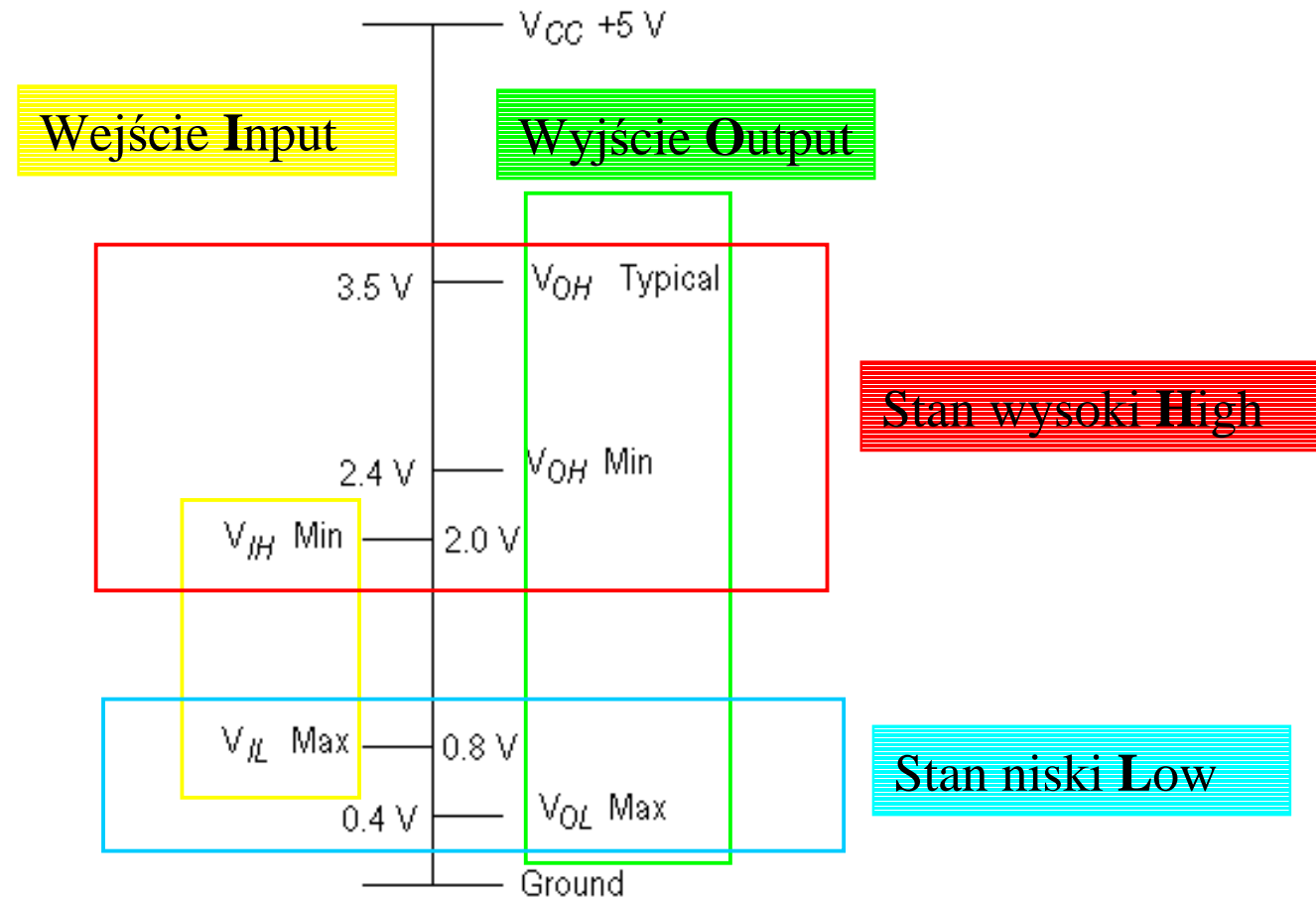


Poziomy napięć w niesymetrycznej linii transmisyjnej w standardzie RS232

# Poziomy napięć stanów logicznych RS232-5V i innych

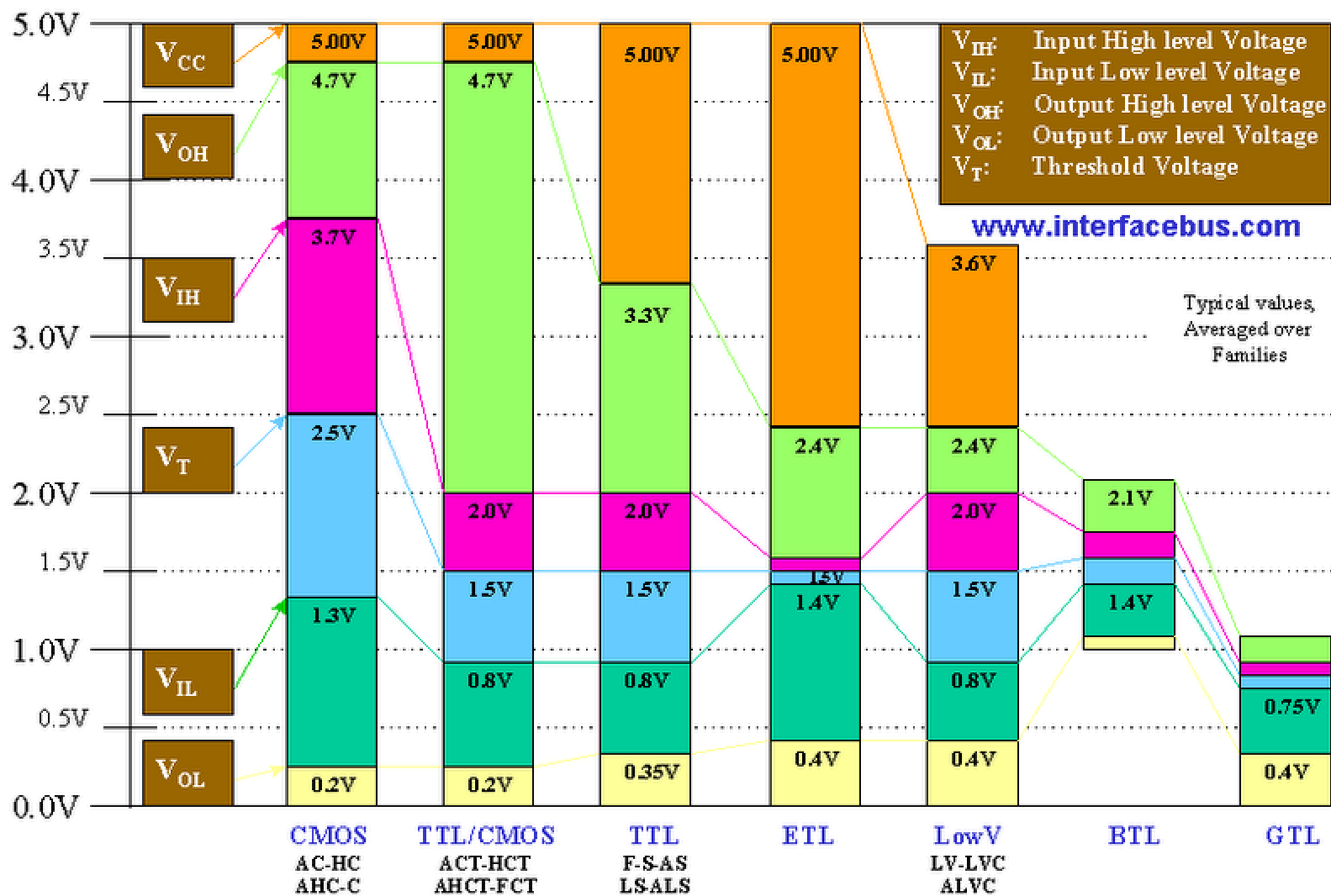


# Poziomy napięć stanów logicznych standardowego TTL

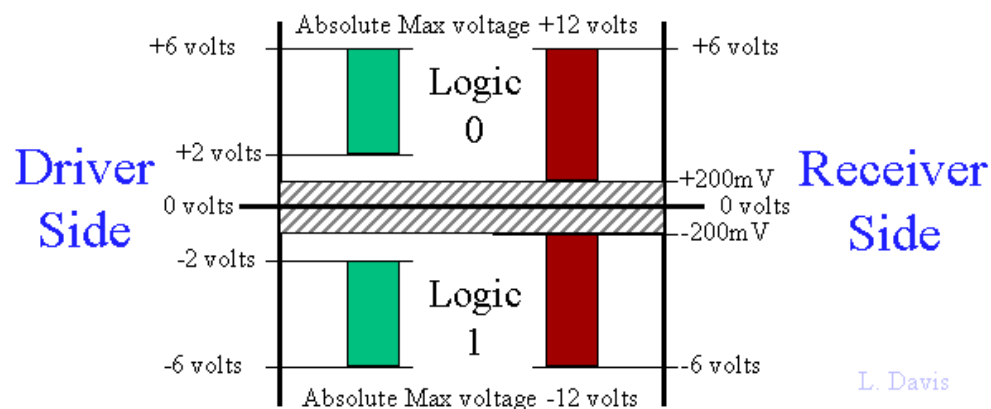
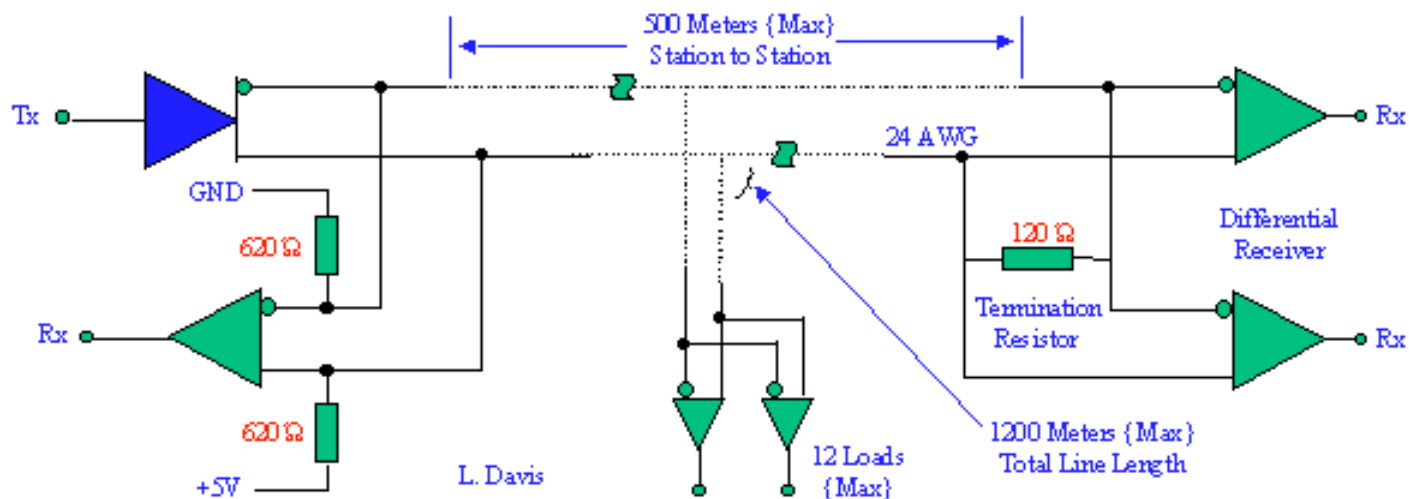


Poziomy napięć w niesymetrycznej linii transmisyjnej w standardzie TTL

## Poziomy napięć stanów logicznych TTL i pochodnych



## Symetryczna linia przesyłowa – przykład RS422



## Konfiguracja symetryczna linii interfejsu 422

## Podstawowe systemy złączy

Najczęściej stosowane rodzaje złączy w systemach interfejsów:

**1 – Złącza szufladowe**

**2 – Złącza krawędziowe**

**3 – Złącza koncentryczne**



## Systemy złączy - szufladowe



Szufladowe Amphenol

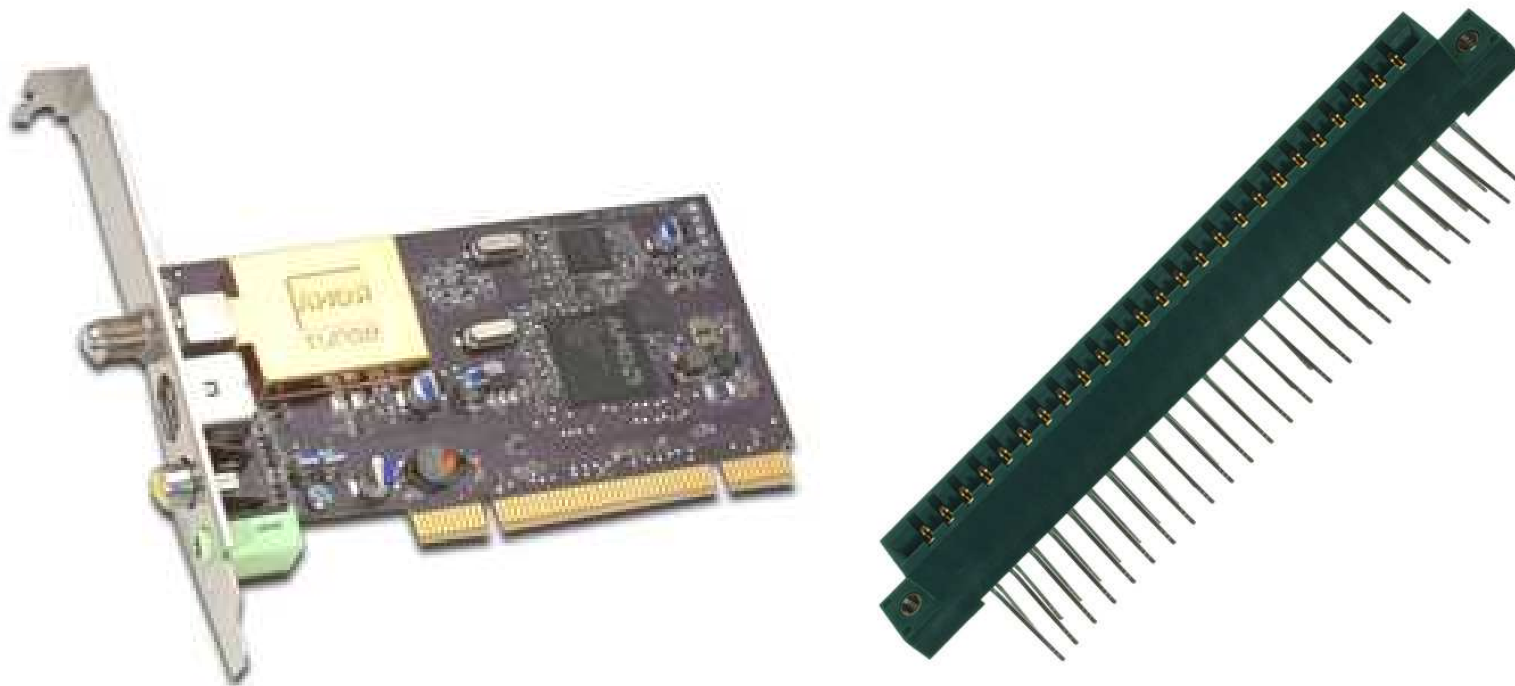


Szufladowe D-SUB Cannon



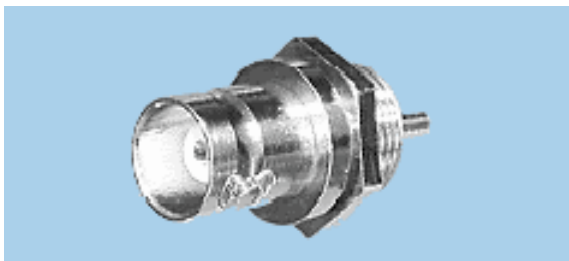
Kabel Centronics

## Systemy złączy - krawędziowe



## Złącze Krawędziowe

## Systemy złączy - koncentryczne



Gniazdo BNC



Wtyczka i trójnik BNC

## Złącze koncentryczne BNC koncentryczne

## Podsumowanie

1. Do podstawowych parametrów opisujących system interfejsu można zaliczyć: zasięg i szybkość transmisji, strukturę logiczną i fizyczną połączeń, sposób transmisji porcji informacji, sposób synchronizacji transmisji, sposób adresowania urządzeń, parametry elektryczne, mechaniczne i okablowanie.
2. Struktura logiczna i fizyczna są ze sobą w pewnym zakresie powiązane.
3. Transmisję danych można zorganizować szeregowo lub równoległe (szeregowo-równoległe), synchronicznie lub asynchronicznie.
4. W logicznej strukturze połączeń innej niż gwiazdzista konieczne jest adresowanie urządzeń.
5. Parametry elektryczne systemu interfejsu przede wszystkim określają poziomy napięcie lub prądów przypisanych logicznym wartościom 0 i 1.
6. Parametry mechaniczne systemu interfejsu opisują przede wszystkim zastosowany system złącz i okablowania.

