Technika cyfrowa i mikroprocesorowa

Transmisja szeregowa, interfejsy komunikacyjne

Wojciech Tarnawski
www.w-tarnawski.pl
wojciech.tarnawski@pwr.edu.pl

Interfejsy komunikacyjne

- Co to jest interfejs komunikacyjny?
- 0/1-bity, bajty, ramki danych, protokoły
- Komunikacja analogowa
- Przewodowe:
  - UART, RS232, RS485
  - 1-wire, I2C, SPI
  - CAN, LIN
  - ETHERNET
  - …
- Bezprzewodowe:
  - 433/868MHz, LoRa
  - WiFi, Bluetooth
  - GSM, GPS
  - …

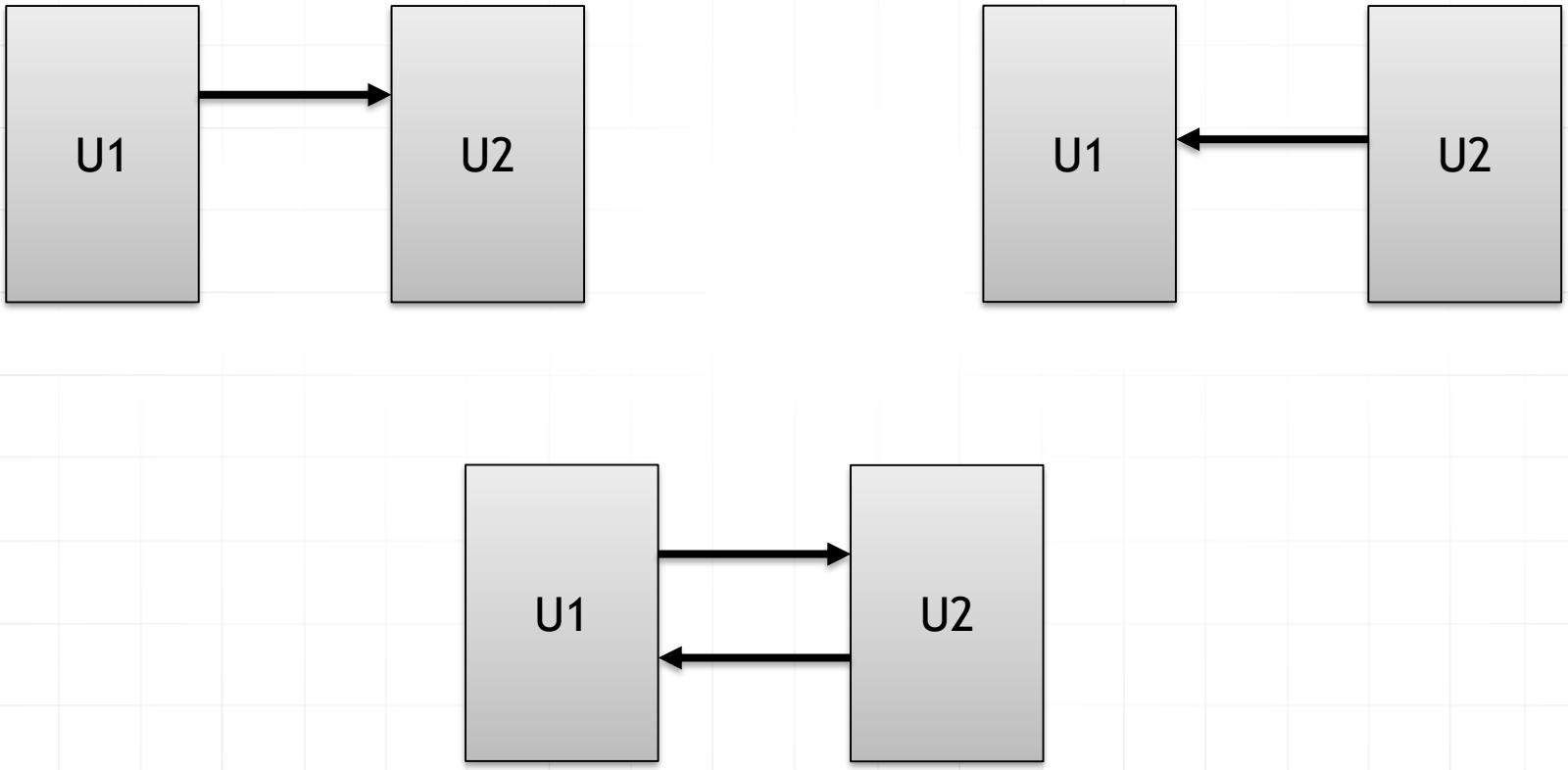
Interfejsy komunikacyjne

Co to jest?

- Interfejs komunikacyjny to elementy umożliwiające wzajemną komunikację (przesyłanie informacji) pomiędzy różnymi urządzeniami
- Interfejs określa warstwę fizyczną jak ilość urządzeń podłączonych i wymieniających dane
- Interfejs zapewnia dopasowanie poziomów elektrycznych sygnałów
- Implementuje sposób przesyłania jak i określa szybkości i format przesyłanych danych.
- Protokół komunikacyjny, wykrywanie błędów transmisji

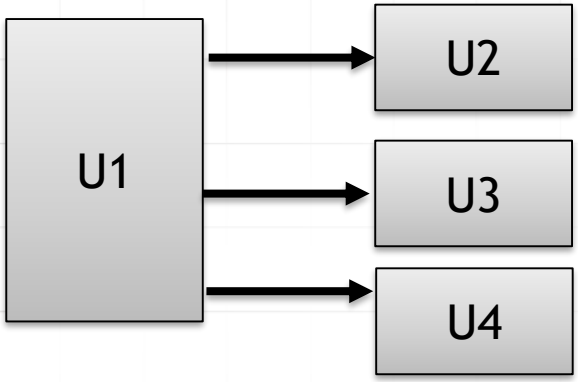
Interfejsy komunikacyjne

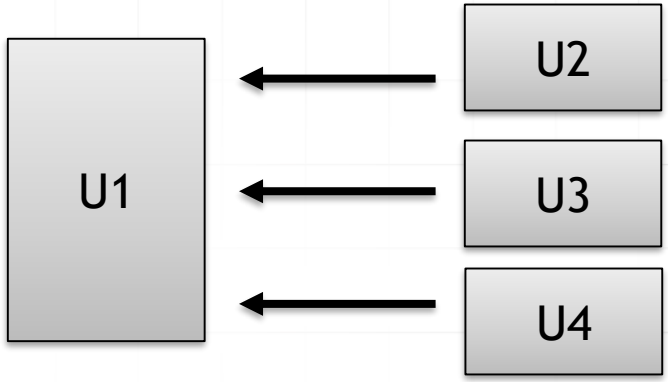
Ilość urządzeń: 1

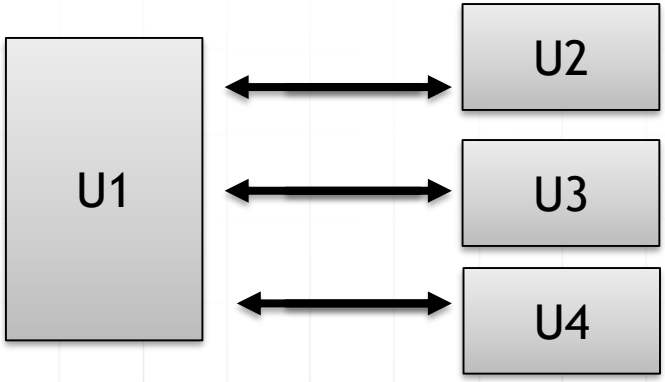


Interfejsy komunikacyjne

Ilość urządzeń: 2 i więcej



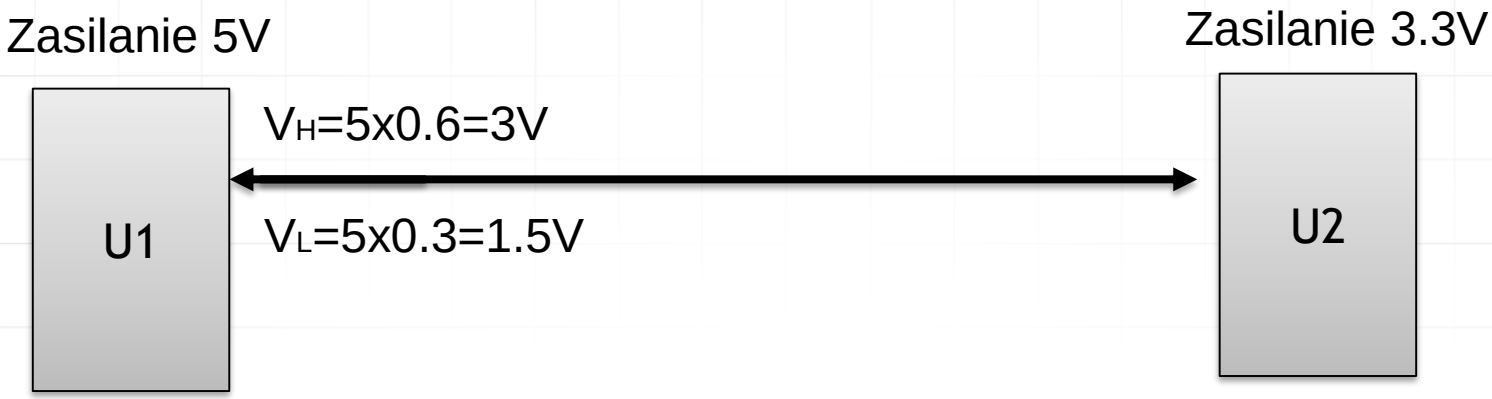




Problem kolizji danych, dostępu do interfejsu

Interfejsy komunikacyjne

Dopasowanie poziomów elektrycznych



Czy możemy podłączyć bezpośrednio?

U1 - Atmega328P

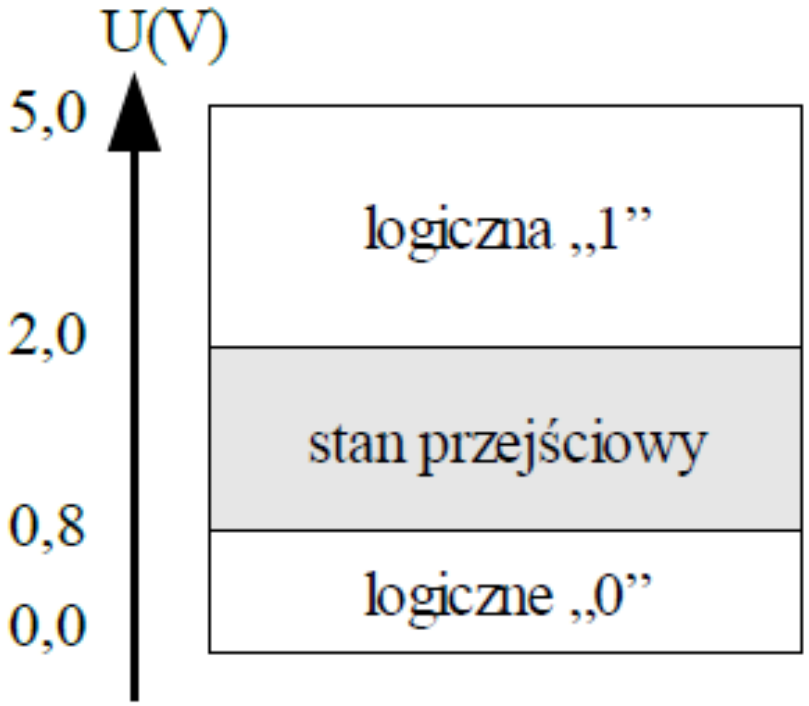
28.2 DC Characteristics

T_A = –40°C to +125°C, V_{CC} = 2.7V to 5.5V (unless otherwise noted)

Parameter	Condition	Symbol	Min.	Typ.	Max.	Unit
Input low voltage, except XTAL1 and RESET pin	V_{CC} = 2.7V to 5.5V	V_{IL}	–0.5		$0.3V_{CC}$ (1)	V
Input high voltage, except XTAL1 and RESET pins	V_{CC} = 2.7V to 5.5V	V_{IH}	$0.6V_{CC}$ (2)		V_{CC} + 0.5	V

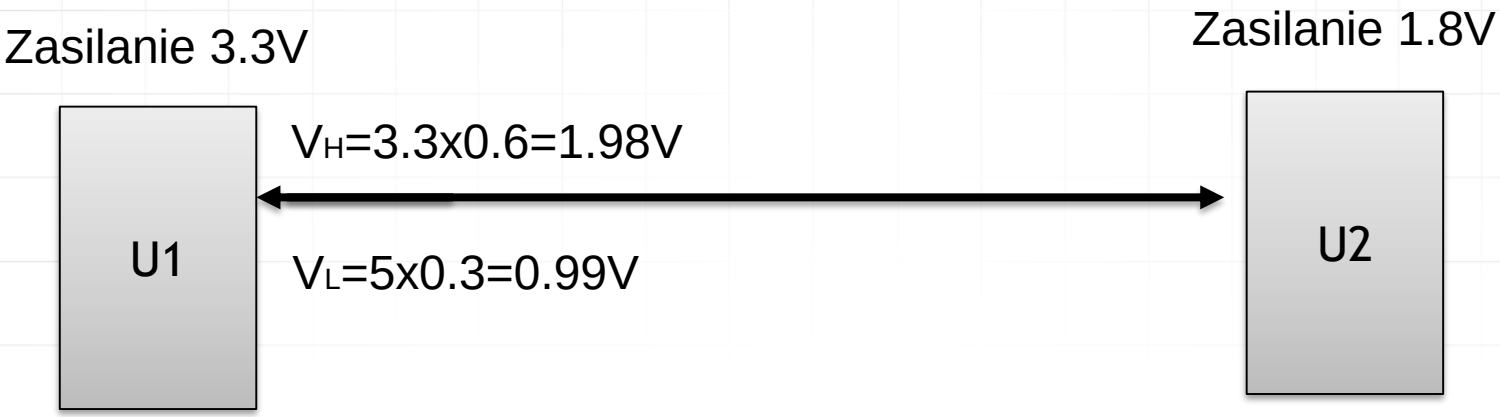
Interfejsy komunikacyjne

Dopasowanie poziomów elektrycznych



Interfejsy komunikacyjne

Dopasowanie poziomów elektrycznych

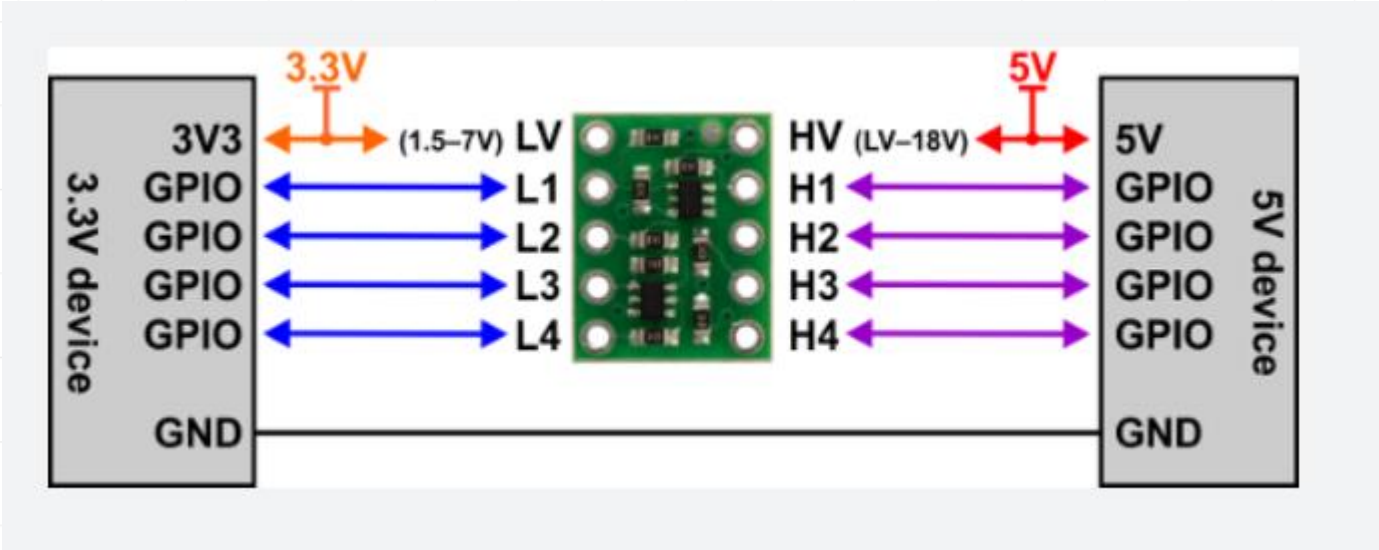


U2 - MPU6050

Parameters	Conditions	Typical	Units	Notes
Primary I^2C I/O (SCL, SDA)				
V_{IL}, LOW-Level Input Voltage	MPU-6000	-0.5 to 0.3*VDD	V	
V_{IH}, HIGH-Level Input Voltage	MPU-6000	0.7*VDD to VDD + 0.5V	V	

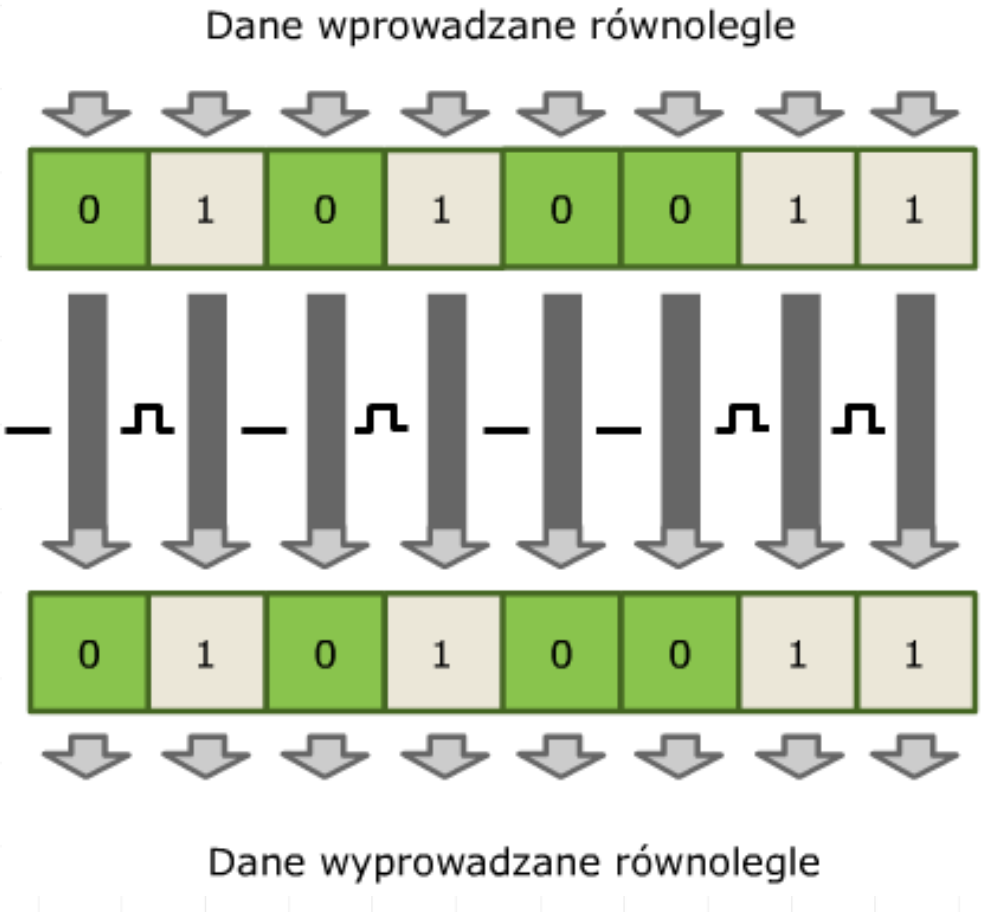
Interfejsy komunikacyjne

Konwerter poziomów logicznych



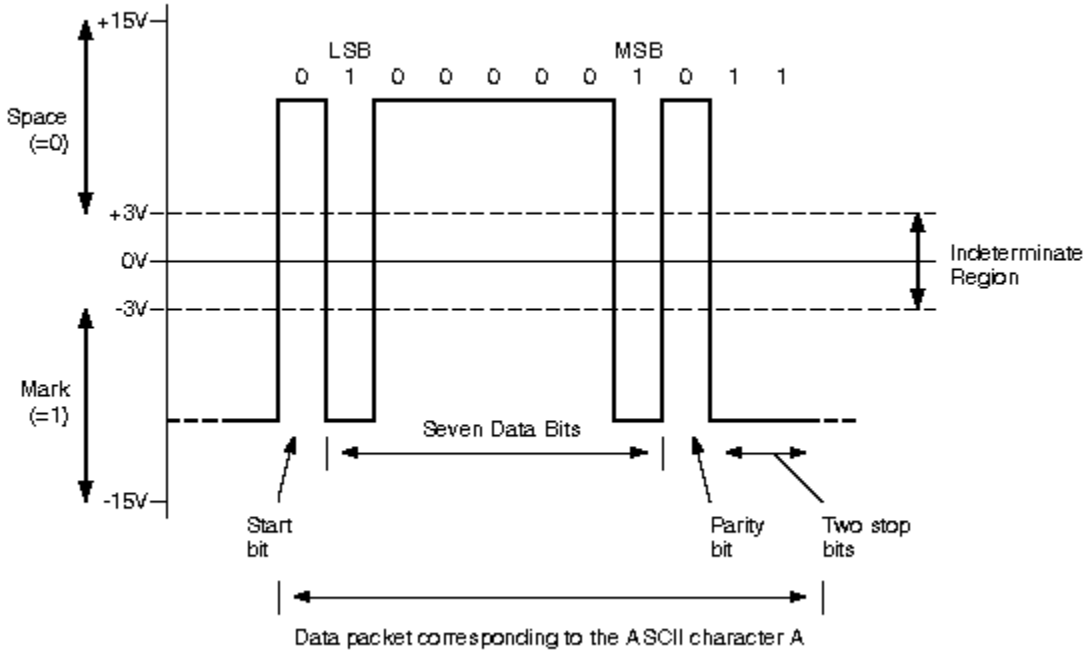
Interfejsy komunikacyjne - przewodowe

Komunikacja cyfrowa równoległa – przykład port LPT

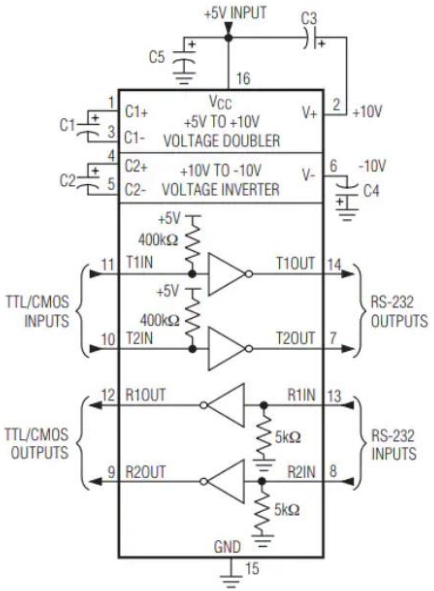


Interfejsy komunikacyjne

Dopasowanie poziomów elektrycznych – RS232

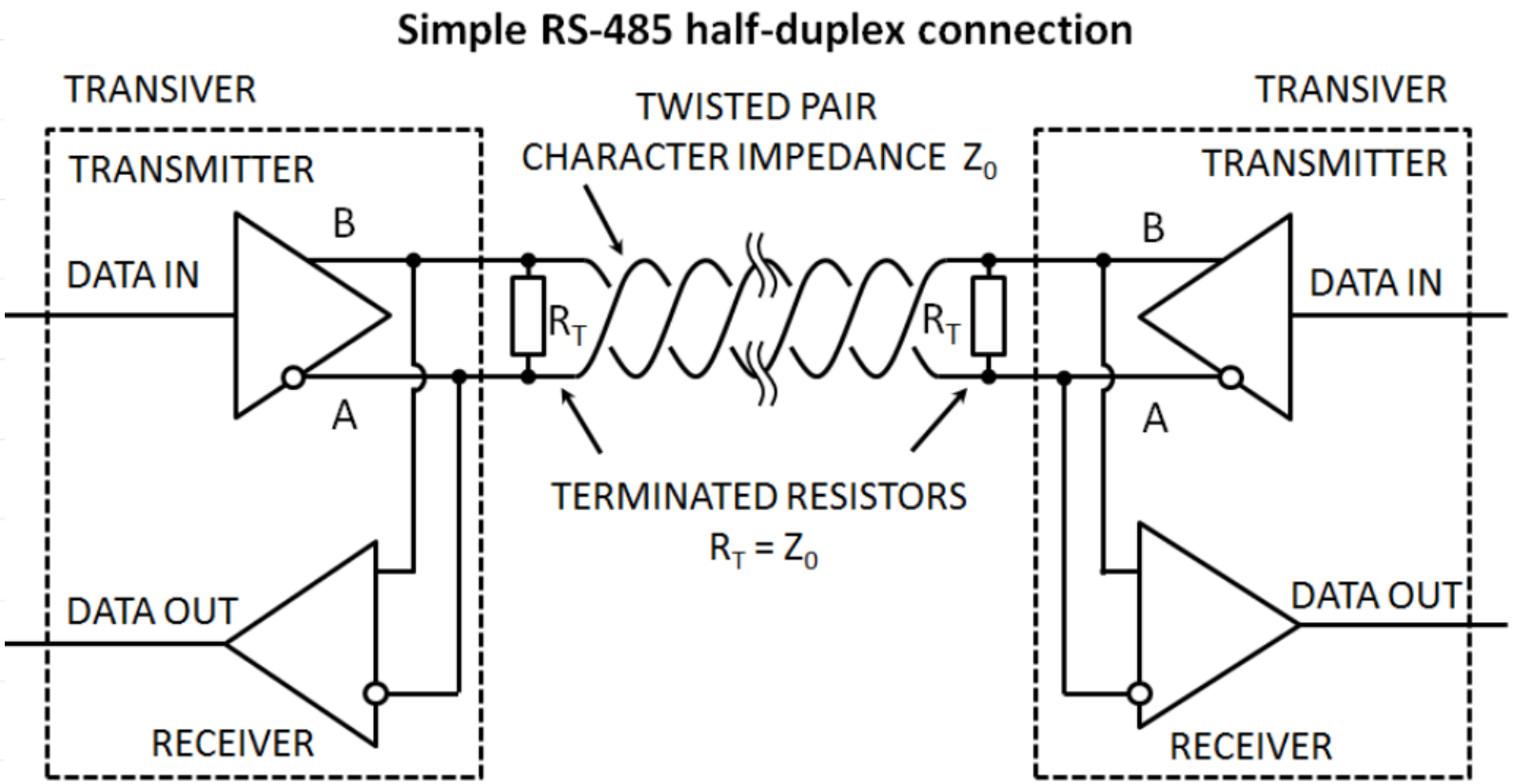


MAX232



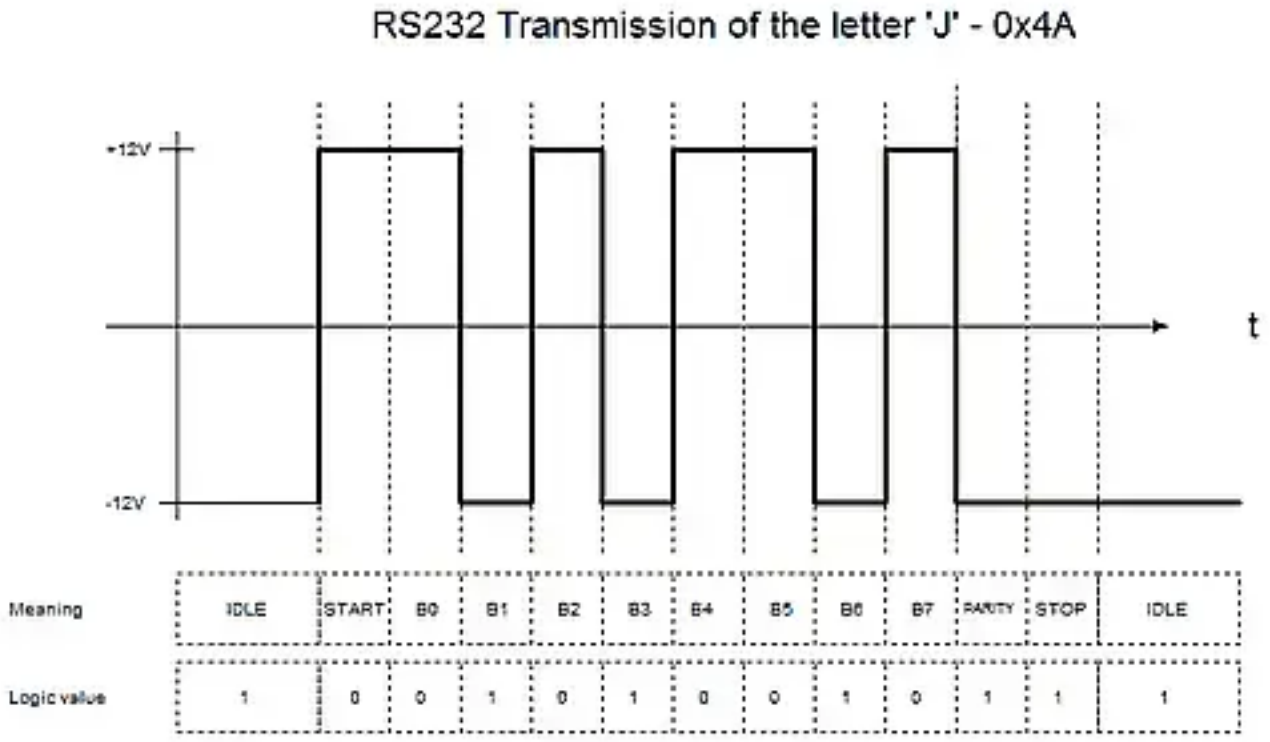
Interfejsy komunikacyjne

Dopasowanie poziomów elektrycznych – RS485 - prąd



Interfejsy komunikacyjne

Stany logiczne na linii – przesłanie 1 bajta - RS232



IDLE START D A N E PARITY STOP IDLE

Interfejsy komunikacyjne

Ramka danych, protokół – MODBUS ASCII

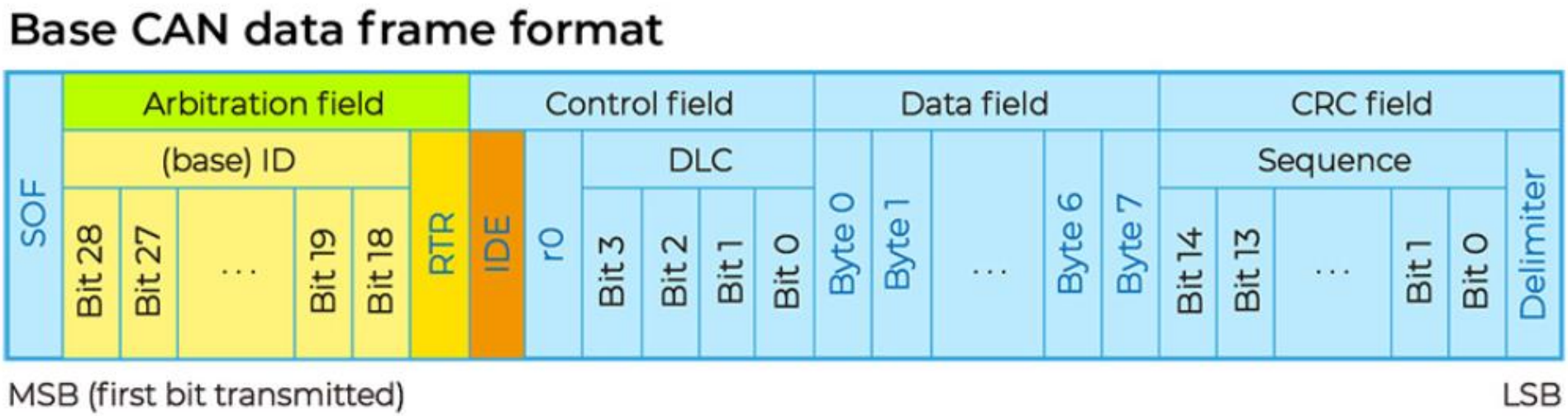
Ramka komunikacji w trybie ASCII

:	Adres		Kod funkcji		Dane			Suma kontrolna		CR	LF
						...					

Potrzeba przesłać 10 bajtów –> wysłać 1 bajt

Interfejsy komunikacyjne

Ramka danych, protokół – CAN ramka podstawowa



potrzeba przesłać n bajtów –> wysłać 1-8 bajów

Interfejsy komunikacyjne - przewodowe

Komunikacja analogowa

- prosta realizacja
- wsparcie przez prawie każdy mikrokontroler-przetwornik A/D
- sygnał ciągły
- prosty „protokół”, przesłanie danych
- 1 linia na jeden komunikat

Interfejsy komunikacyjne - przewodowe

Komunikacja analogowa – czujnik temperatury TMP36GT9Z

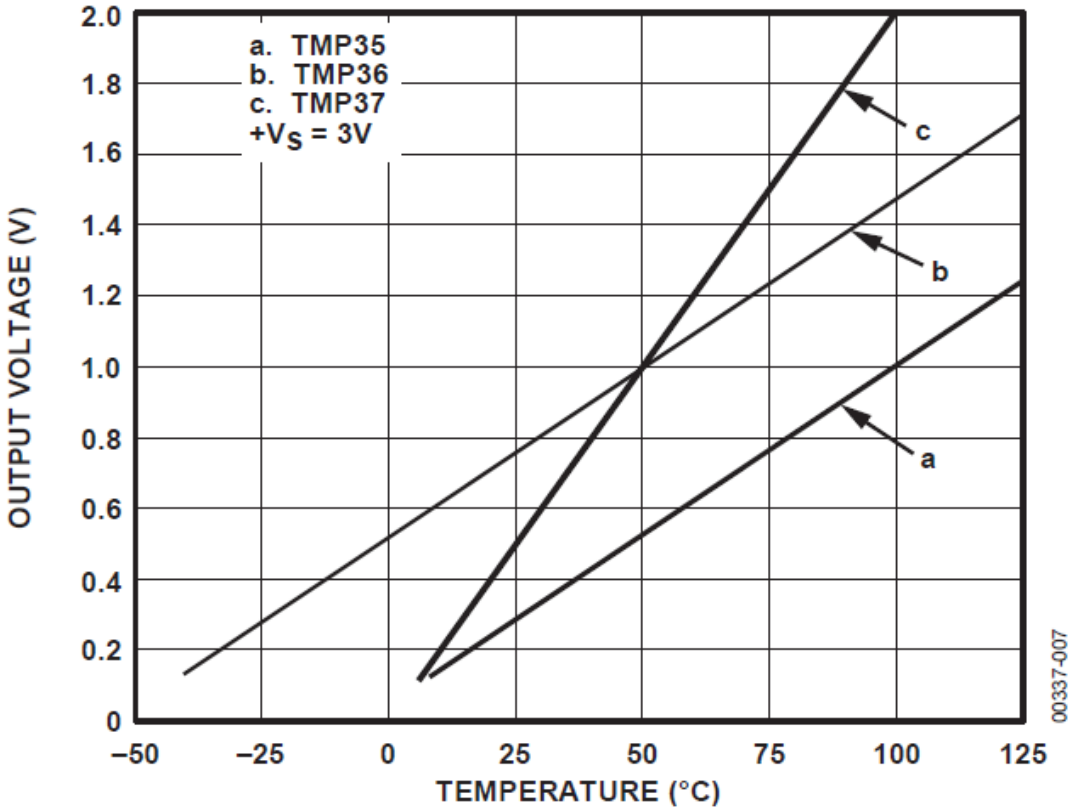
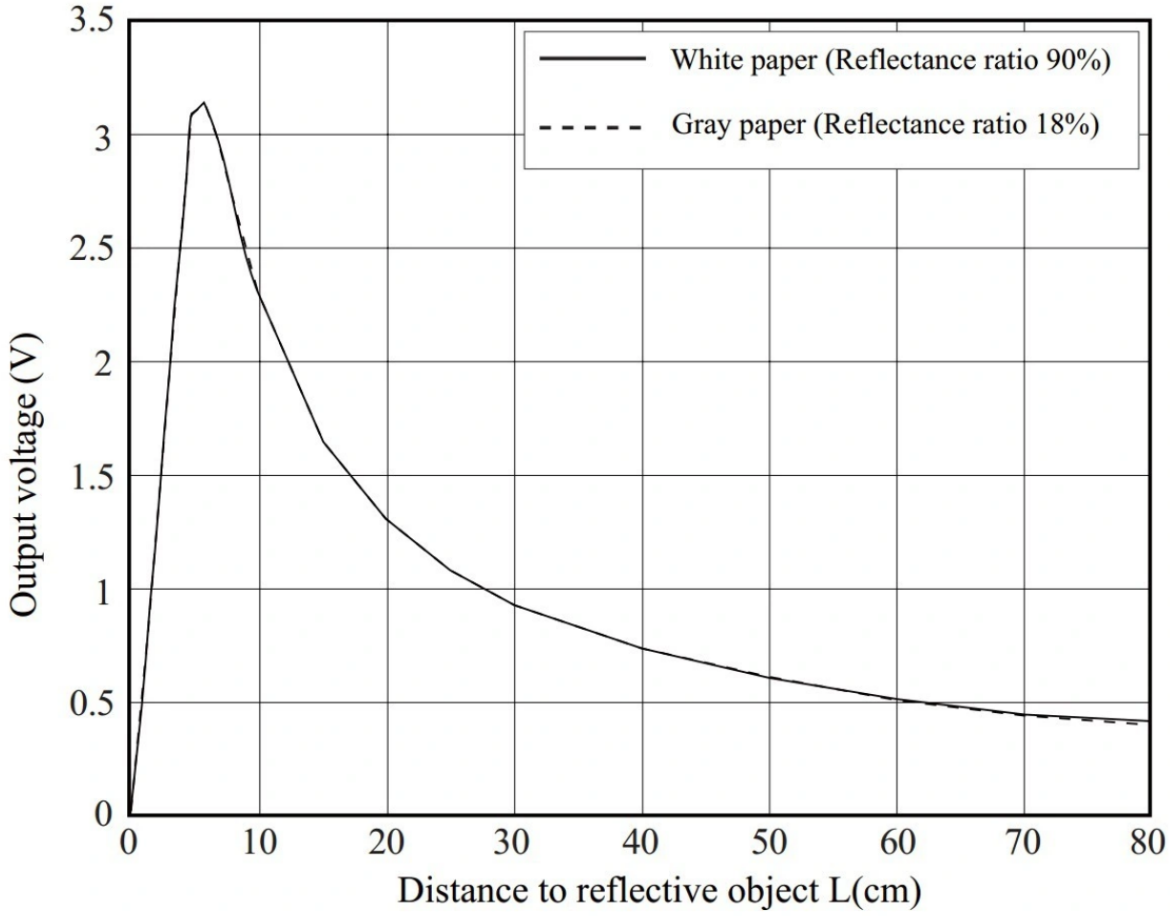


Figure 6. Output Voltage vs. Temperature

Interfejsy komunikacyjne - przewodowe

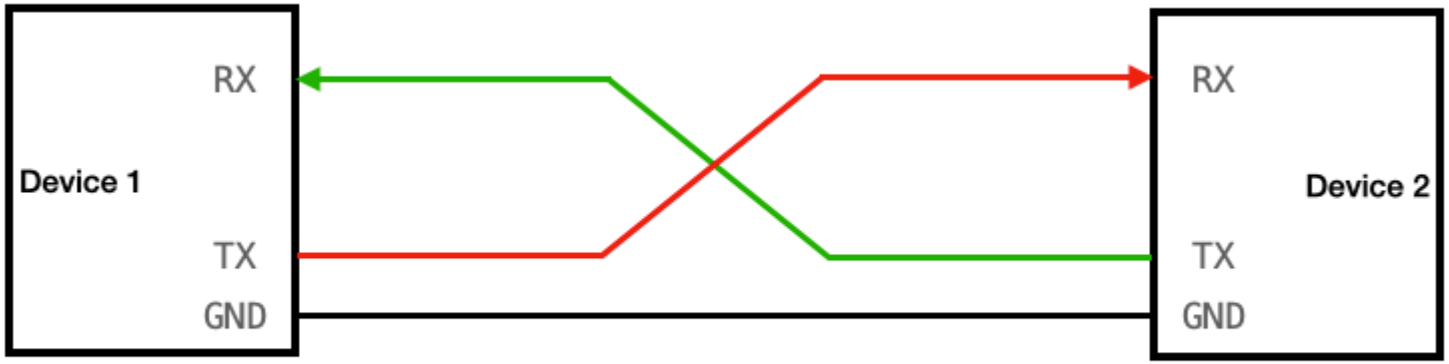
Komunikacja analogowa – czujnik odległości GP2Y0A21YK0F



Interfejsy komunikacyjne - przewodowe

UART/ USART

- 2 przewody – Tx – tor nadawania danych, Rx – tor odbierania danych
- UART asynchroniczny – brak sygnału zegarowego
- USART synchroniczny – dodatkowy sygnał taktujący - XCK
- dwa urządzenia – transmisja dwukierunkowa
- konieczność ustawienia parametrów transmisji – 9600,8,n,1
- prosty interfejs, bardzo popularny



Interfejsy komunikacyjne - przewodowe

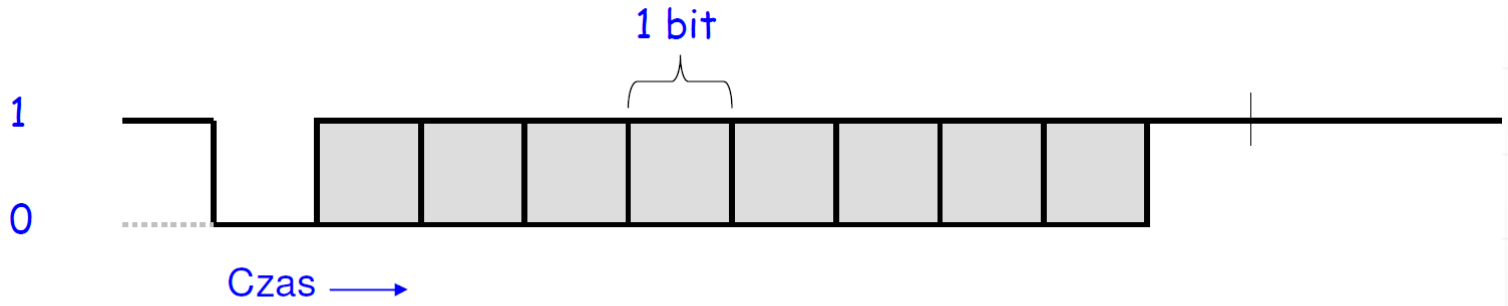
UART – czas przesłania bitu - baudrate

Prędkość: 9600bps (bitów na sekundę),
czas przesłania jednego bitu: 1/9600~104,16us

Interfejsy komunikacyjne - przewodowe

UART – przesłanie bajtu – baudrate

Prędkość: 9600bps (bitów na sekundę), długość jednego bitu: 104,16us

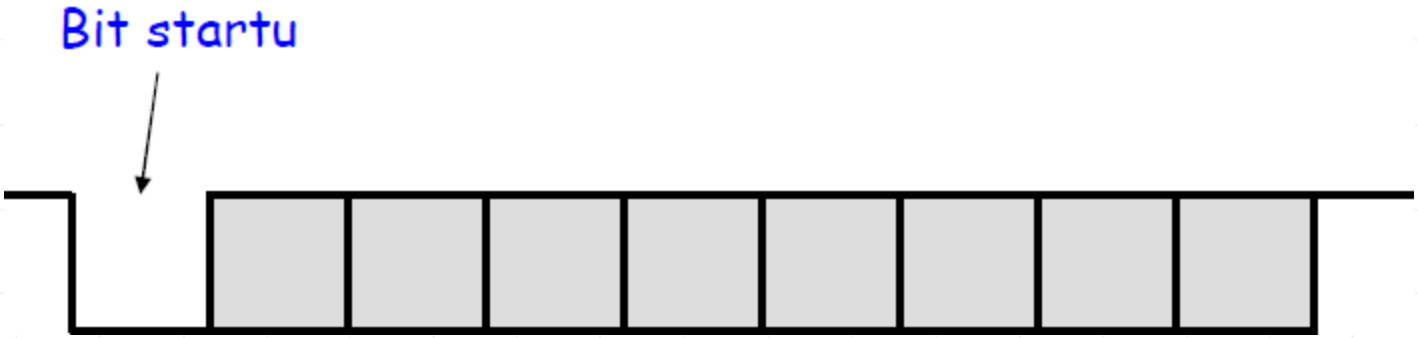


Interfejsy komunikacyjne - przewodowe

UART – przesłanie bajtu – bit startu

Prędkość: 9600bps (bitów na sekundę), długość jednego bitu: 104,16us

- bit startu rozpoczyna transmisję
- wartość tego bitu zawsze wynosi 0
- zbocze opadające jest sygnałem dla odbiornika do synchronizacji odbioru danych

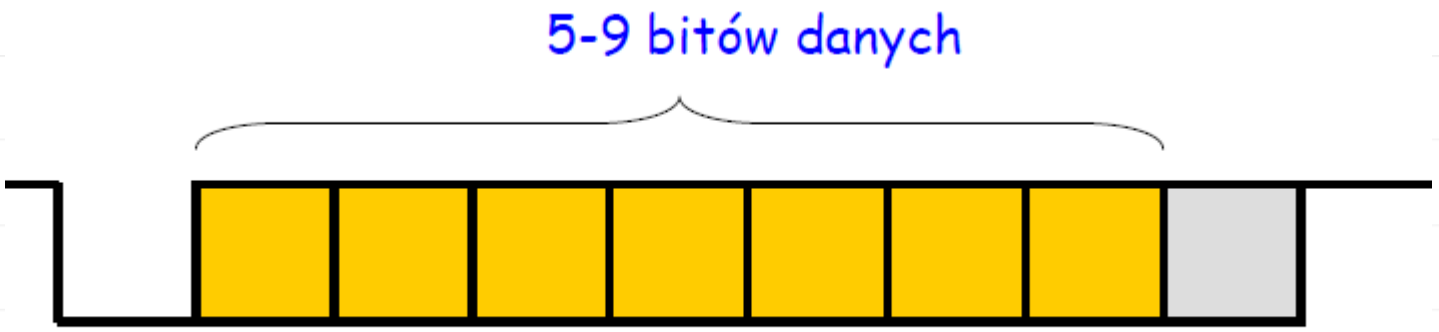


Interfejsy komunikacyjne - przewodowe

UART – przesłanie bajtu – dane

Prędkość: 9600bps (bitów na sekundę), długość jednego bitu: 104,16us

- po bicie startu wysyłane są dane
- 5-9 bitów

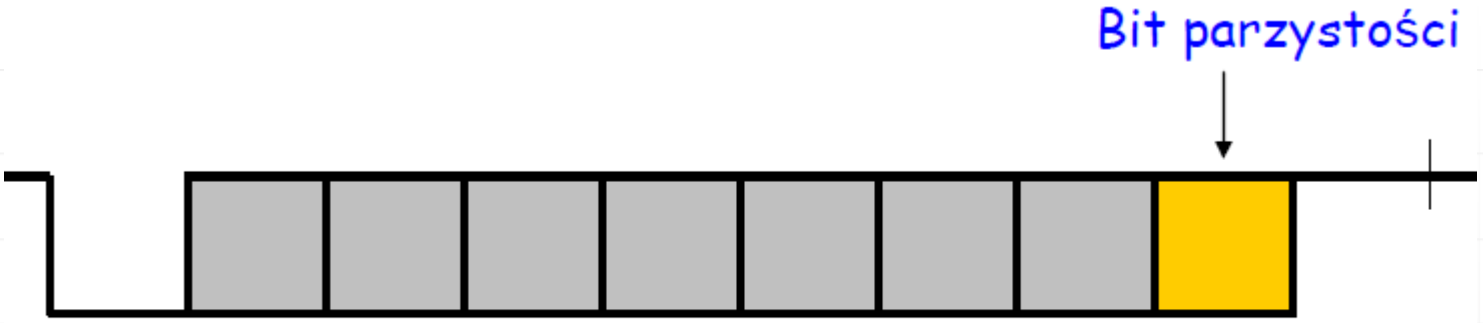


Interfejsy komunikacyjne - przewodowe

UART – przesłanie bajtu – bit parzystości

Prędkość: 9600bps (bitów na sekundę), długość jednego bitu: 104,16us

- po wysłaniu danych może nastąpić opcjonalny bit parzystości
- wysłanie zależne od konfiguracji
- proste zabezpieczenie na błędy transmisji danych

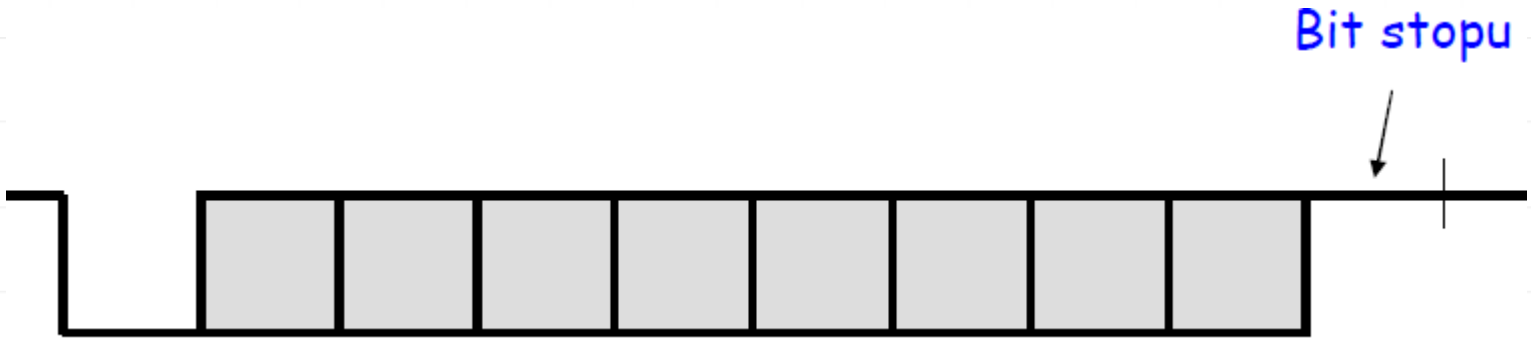


Interfejsy komunikacyjne - przewodowe

UART – przesłanie bajtu – bit stopu

Prędkość: 9600bps (bitów na sekundę), długość jednego bitu: 104,16us

- transmisja kończy się 1-2 bitami stopu
- wysłanie zależne od konfiguracji
- bit stopu ma wartość 1

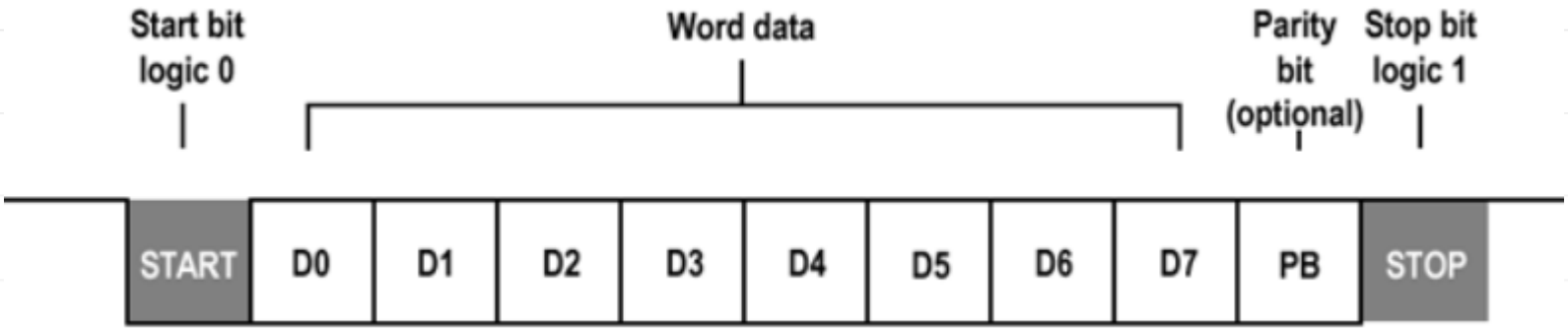


Interfejsy komunikacyjne - przewodowe

UART – przesłanie bajtu

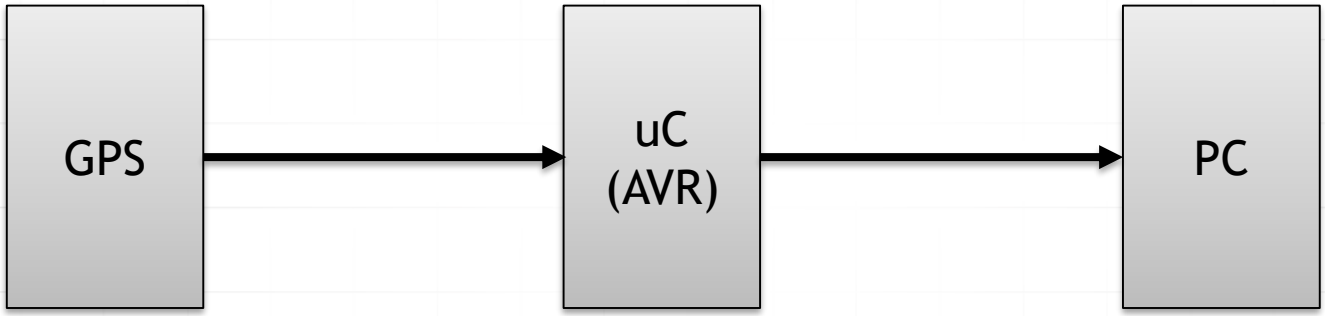
Najczęściej stosowana jest ramka 8,n,1
(8 bitów danych, bez bitu parzystości, 1 bit stopu).

Typowe prędkości: 2400, 4800, 9600, 19200, 38400, 57600, 115200



Interfejsy komunikacyjne - przewodowe

UART – 3 urządzenia???



Interfejsy komunikacyjne

UART – ATMEGA328P

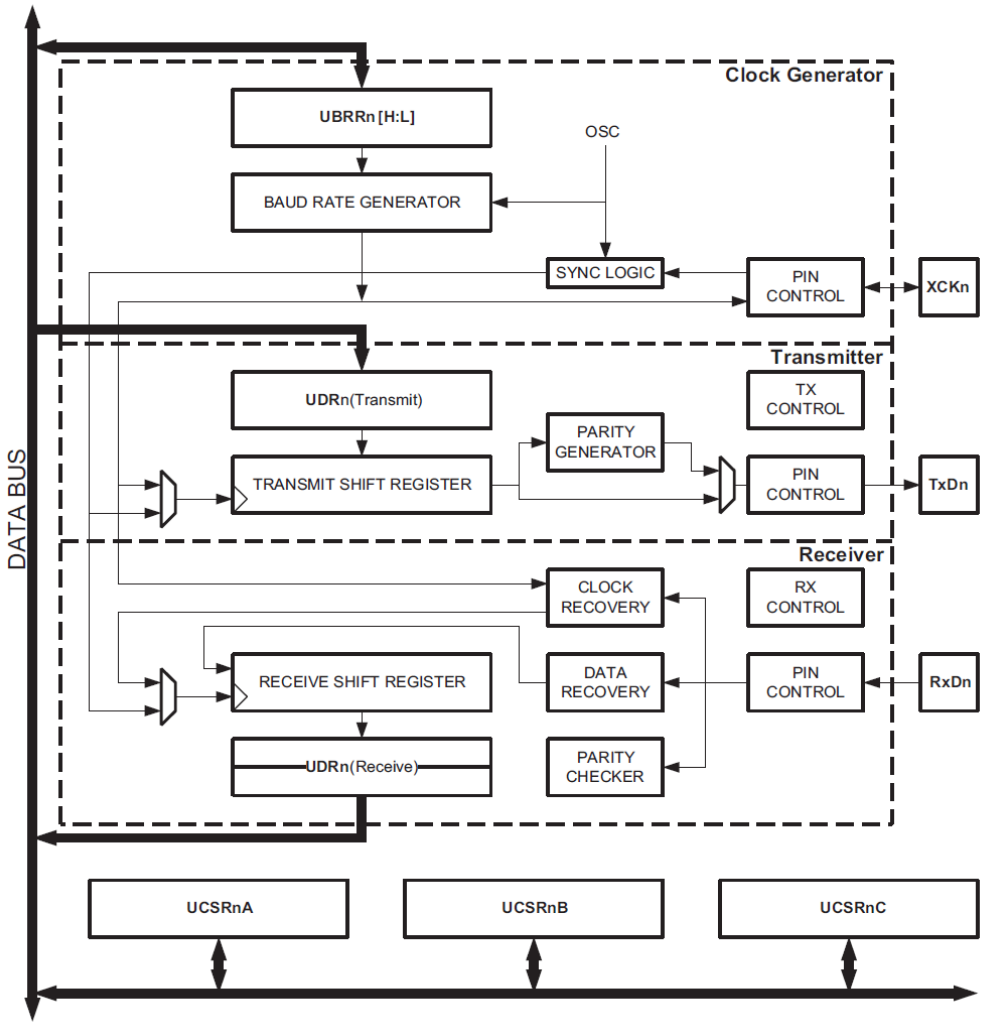
20. USART0

20.1 Features

- Full Duplex Operation (Independent Serial Receive and Transmit Registers)
- Asynchronous or Synchronous Operation
- Master or Slave Clocked Synchronous Operation
- High Resolution Baud Rate Generator
- Supports Serial Frames with 5, 6, 7, 8, or 9 Data Bits and 1 or 2 Stop Bits
- Odd or Even Parity Generation and Parity Check Supported by Hardware
- Data OverRun Detection
- Framing Error Detection
- Noise Filtering Includes False Start Bit Detection and Digital Low Pass Filter
- Three Separate Interrupts on TX Complete, TX Data Register Empty and RX Complete
- Multi-processor Communication Mode
- Double Speed Asynchronous Communication Mode

Interfejsy komunikacyjne

UART – ATMEGA328P



Interfejsy komunikacyjne

UART – ATMEGA328P

$$BAUD = \frac{f_{OSC}}{16(UBRRn + 1)}$$

20.11.5 UBRRnL and UBRRnH – USART Baud Rate Registers

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	UBRRn[11:8]				UBRRnH
	UBRRn[7:0]								UBRRnL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

- **Bit 15:12 – Reserved**

These bits are reserved for future use. For compatibility with future devices, these bit must be written to zero when UBRRnH is written.

- **Bit 11:0 – UBRR[11:0]: USART Baud Rate Register**

This is a 12-bit register which contains the USART baud rate. The UBRRnH contains the four most significant bits, and the UBRRnL contains the eight least significant bits of the USART baud rate. Ongoing transmissions by the Transmitter and Receiver will be corrupted if the baud rate is changed. Writing UBRRnL will trigger an immediate update of the baud rate prescaler.

$$Baud = 9600$$
$$F_{osc} = 16\,000\,000$$
$$UBRRn = ?$$

Interfejsy komunikacyjne

UART – ATMEGA328P

Table 20-7. Examples of UBRRn Settings for Commonly Used Oscillator Frequencies (Continued)

Baud Rate (bps)	f_{osc} = 16.0000MHz				f_{osc} = 18.4320MHz				f_{osc} = 20.0000MHz			
	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error
2400	416	-0.1%	832	0.0%	479	0.0%	959	0.0%	520	0.0%	1041	0.0%
4800	207	0.2%	416	-0.1%	239	0.0%	479	0.0%	259	0.2%	520	0.0%
9600	103	0.2%	207	0.2%	119	0.0%	239	0.0%	129	0.2%	259	0.2%
14.4k	68	0.6%	138	-0.1%	79	0.0%	159	0.0%	86	-0.2%	173	-0.2%
19.2k	51	0.2%	103	0.2%	59	0.0%	119	0.0%	64	0.2%	129	0.2%
28.8k	34	-0.8%	68	0.6%	39	0.0%	79	0.0%	42	0.9%	86	-0.2%
38.4k	25	0.2%	51	0.2%	29	0.0%	59	0.0%	32	-1.4%	64	0.2%
57.6k	16	2.1%	34	-0.8%	19	0.0%	39	0.0%	21	-1.4%	42	0.9%
76.8k	12	0.2%	25	0.2%	14	0.0%	29	0.0%	15	1.7%	32	-1.4%
115.2k	8	-3.5%	16	2.1%	9	0.0%	19	0.0%	10	-1.4%	21	-1.4%
230.4k	3	8.5%	8	-3.5%	4	0.0%	9	0.0%	4	8.5%	10	-1.4%
250k	3	0.0%	7	0.0%	4	-7.8%	8	2.4%	4	0.0%	9	0.0%
0.5M	1	0.0%	3	0.0%	–	–	4	-7.8%	–	–	4	0.0%
1M	0	0.0%	1	0.0%	–	–	–	–	–	–	–	–
Max. (1)	1Mbps		2Mbps		1.152Mbps		2.304Mbps		1.25Mbps		2.5Mbps	

1. UBRRn = 0, Error = 0.0%

Interfejsy komunikacyjne

UART – ATMEGA328P

20.11.3 UCSRnB – USART Control and Status Register n B

Bit	7	6	5	4	3	2	1	0	
	RXCIEn	TXCIEn	UDRIEn	RXENn	TXENn	UCSZn2	RXB8n	TXB8n	UCSRnB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – RXCIEn: RX Complete Interrupt Enable n**

Writing this bit to one enables interrupt on the RXCn Flag. A USART Receive Complete interrupt will be generated only if the RXCIEn bit is written to one, the Global Interrupt Flag in SREG is written to one and the RXCn bit in UCSRnA is set.

- **Bit 6 – TXCIEn: TX Complete Interrupt Enable n**

Writing this bit to one enables interrupt on the TXCn Flag. A USART Transmit Complete interrupt will be generated only if the TXCIEn bit is written to one, the Global Interrupt Flag in SREG is written to one and the TXCn bit in UCSRnA is set.

Interfejsy komunikacyjne

UART – ATMEGA328P

20.11.3 UCSRnB – USART Control and Status Register n B

Bit	7	6	5	4	3	2	1	0	
	RXCIEn	TXCIEn	UDRIEn	RXENn	TXENn	UCSZn2	RXB8n	TXB8n	UCSRnB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 5 – UDRIEn: USART Data Register Empty Interrupt Enable n**

Writing this bit to one enables interrupt on the UDREn Flag. A Data Register Empty interrupt will be generated only if the UDRIEn bit is written to one, the Global Interrupt Flag in SREG is written to one and the UDREn bit in UCSRnA is set.

- **Bit 4 – RXENn: Receiver Enable n**

Writing this bit to one enables the USART Receiver. The Receiver will override normal port operation for the RxDn pin when enabled. Disabling the Receiver will flush the receive buffer invalidating the FEn, DORn, and UPEn Flags.

- **Bit 3 – TXENn: Transmitter Enable n**

Writing this bit to one enables the USART Transmitter. The Transmitter will override normal port operation for the TxDn pin when enabled. The disabling of the Transmitter (writing TXENn to zero) will not become effective until ongoing and pending transmissions are completed, i.e., when the Transmit Shift Register and Transmit Buffer Register do not contain data to be transmitted. When disabled, the Transmitter will no longer override the TxDn port.

Interfejsy komunikacyjne

UART – ATMEGA328P

20.11.4 UCSRnC – USART Control and Status Register n C

Bit	7	6	5	4	3	2	1	0	
	UMSELn1	UMSELn0	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	UCSRnC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	1	1	0	

- **Bits 7:6 – UMSELn1:0 USART Mode Select**

These bits select the mode of operation of the USARTn as shown in Table 20-8.

Table 20-8. UMSELn Bits Settings

UMSELn1	UMSELn0	Mode
0	0	Asynchronous USART
0	1	Synchronous USART
1	0	(Reserved)
1	1	Master SPI (MSPIM)(1)

Interfejsy komunikacyjne

UART – ATMEGA328P

20.11.4 UCSRnC – USART Control and Status Register n C

Bit	7	6	5	4	3	2	1	0	
	UMSELn1	UMSELn0	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	UCSRnC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	1	1	0	

- **Bits 5:4 – UPMn1:0: Parity Mode**

These bits enable and set type of parity generation and check. If enabled, the Transmitter will automatically generate and send the parity of the transmitted data bits within each frame. The Receiver will generate a parity value for the incoming data and compare it to the UPMn setting. If a mismatch is detected, the UPEn Flag in UCSRnA will be set.

Table 20-9. UPMn Bits Settings

UPMn1	UPMn0	Parity Mode
0	0	Disabled
0	1	Reserved
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity

- **Bit 3 – USBSn: Stop Bit Select**

This bit selects the number of stop bits to be inserted by the Transmitter. The Receiver ignores this setting.

Table 20-10. USBS Bit Settings

USBSn	Stop Bit(s)
0	1-bit
1	2-bit

Interfejsy komunikacyjne

UART – ATMEGA328P

20.11.4 UCSRnC – USART Control and Status Register n C

Bit	7	6	5	4	3	2	1	0	
	UMSELn1	UMSELn0	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	UCSRnC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	1	1	0	

- **Bit 2:1 – UCSZn1:0: Character Size**

The UCSZn1:0 bits combined with the UCSZn2 bit in UCSRnB sets the number of data bits (Character SiZe) in a frame the Receiver and Transmitter use.

Table 20-11. UCSZn Bits Settings

UCSZn2	UCSZn1	UCSZn0	Character Size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9-bit

Interfejsy komunikacyjne

UART – ATMEGA328P

20.11.4 UCSRnC – USART Control and Status Register n C

Bit	7	6	5	4	3	2	1	0	
	UMSELn1	UMSELn0	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	UCSRnC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	1	1	0	

- **Bit 2:1 – UCSZn1:0: Character Size**

The UCSZn1:0 bits combined with the UCSZn2 bit in UCSRnB sets the number of data bits (Character SiZe) in a frame the Receiver and Transmitter use.

Table 20-11. UCSZn Bits Settings

UCSZn2	UCSZn1	UCSZn0	Character Size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9-bit

Interfejsy komunikacyjne

UART – ATMEGA328P

Bit	7	6	5	4	3	2	1	0	
	RXCn	TXCn	UDREn	FEn	DORn	UPEn	U2Xn	MPCMn	UCSRnA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

- **Bit 7 – RXCn: USART Receive Complete**

This flag bit is set when there are unread data in the receive buffer and cleared when the receive buffer is empty (i.e., does not contain any unread data). If the Receiver is disabled, the receive buffer will be flushed and consequently the RXCn bit will become zero. The RXCn Flag can be used to generate a Receive Complete interrupt (see description of the RXCIEn bit).

- **Bit 6 – TXCn: USART Transmit Complete**

This flag bit is set when the entire frame in the Transmit Shift Register has been shifted out and there are no new data currently present in the transmit buffer (UDRn). The TXCn Flag bit is automatically cleared when a transmit complete interrupt is executed, or it can be cleared by writing a one to its bit location. The TXCn Flag can generate a Transmit Complete interrupt (see description of the TXCIEn bit).

- **Bit 5 – UDREn: USART Data Register Empty**

The UDREn Flag indicates if the transmit buffer (UDRn) is ready to receive new data. If UDREn is one, the buffer is empty, and therefore ready to be written. The UDREn Flag can generate a Data Register Empty interrupt (see description of the UDRIEn bit). UDREn is set after a reset to indicate that the Transmitter is ready.

Interfejsy komunikacyjne

UART – ATMEGA328P

20.11.1 UDRn – USART I/O Data Register n

Bit	7	6	5	4	3	2	1	0	
	RXB[7:0]								UDRn (Read)
	TXB[7:0]								UDRn (Write)
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The USART Transmit Data Buffer Register and USART Receive Data Buffer Registers share the same I/O address referred to as USART Data Register or UDRn. The Transmit Data Buffer Register (TXB) will be the destination for data written to the UDRn Register location. Reading the UDRn Register location will return the contents of the Receive Data Buffer Register (RXB).

For 5-, 6-, or 7-bit characters the upper unused bits will be ignored by the Transmitter and set to zero by the Receiver.

The transmit buffer can only be written when the UDREn Flag in the UCSRnA Register is set. Data written to UDRn when the UDREn Flag is not set, will be ignored by the USART Transmitter. When data is written to the transmit buffer, and the Transmitter is enabled, the Transmitter will load the data into the Transmit Shift Register when the Shift Register is empty. Then the data will be serially transmitted on the TxDn pin.

The receive buffer consists of a two level FIFO. The FIFO will change its state whenever the receive buffer is accessed. Due to this behavior of the receive buffer, do not use Read-Modify-Write instructions (SBI and CBI) on this location. Be careful when using bit test instructions (SBIC and SBIS), since these also will change the state of the FIFO.

Interfejsy komunikacyjne

UART – ATMEGA328P

C Code Example[1]

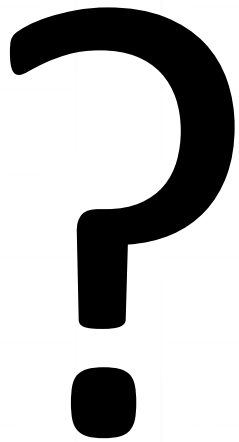
```
unsigned char USART_Receive( void )
{
  /* Wait for data to be received */
  while ( !(UCSRnA & (1<<RXCn)) )
        ;
  /* Get and return received data from buffer */
  return UDRn;
}
```

Interfejsy komunikacyjne

UART – ATMEGA328P

C Code Example[1]

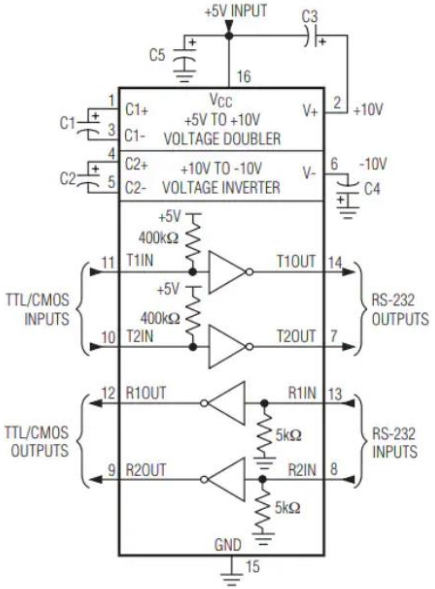
```
void USART_Transmit( unsigned char data )
{
  /* Wait for empty transmit buffer */
  while ( !( UCSRnA & (1<<UDREn)) )
        ;
  /* Put data into buffer, sends the data */
  UDRn = data;
}
```

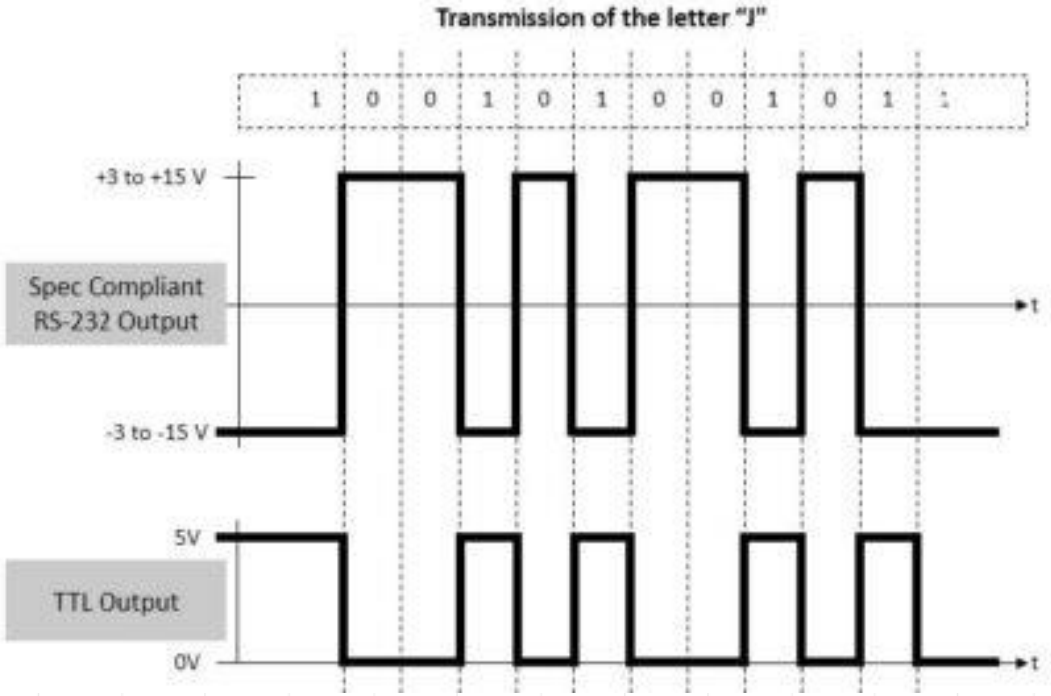


Interfejsy komunikacyjne

UART – > RS232

MAX232





Interfejsy komunikacyjne

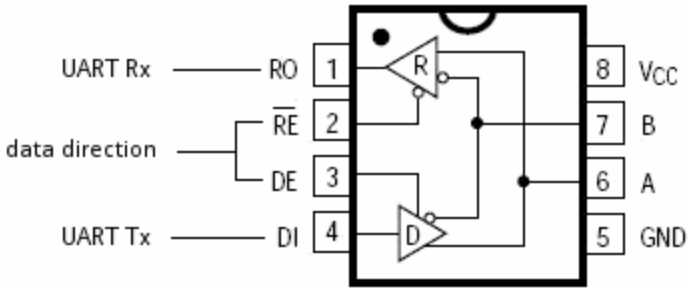
UART – > RS232

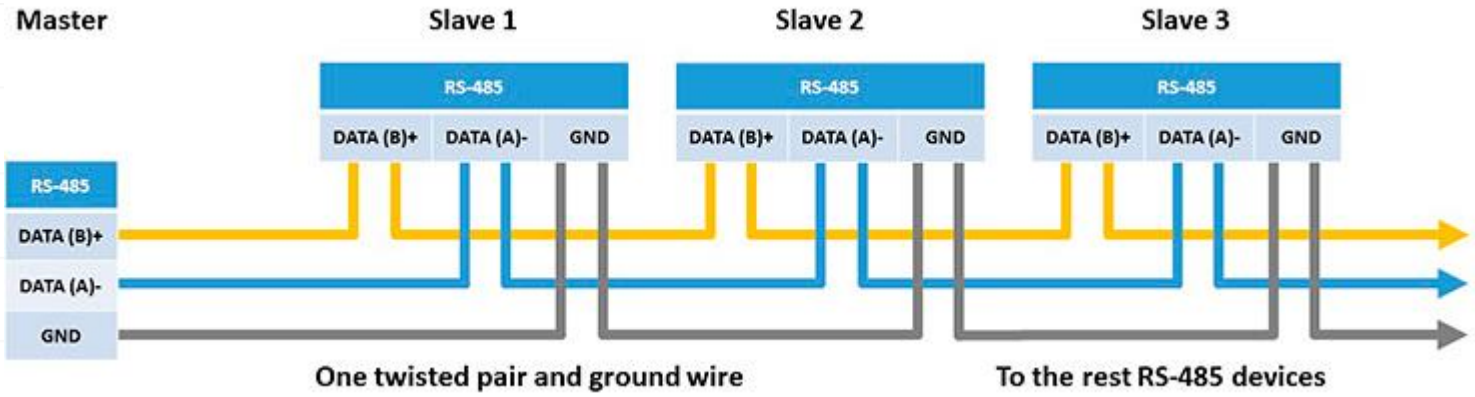
- dwa przewody – prosta budowa, niski koszt
- łączy tylko dla urządzenia
- dobra odporność na zakłócenia zewnętrzne + zabezpieczenia wbudowane w MAX232
- długość do 15m
- niska prędkość transmisji
- kiedyś jeden z podstawowych interfejsów komputerowych, obecnie wyparty przez USB

Interfejsy komunikacyjne

UART – > RS485

MAX485





Interfejsy komunikacyjne

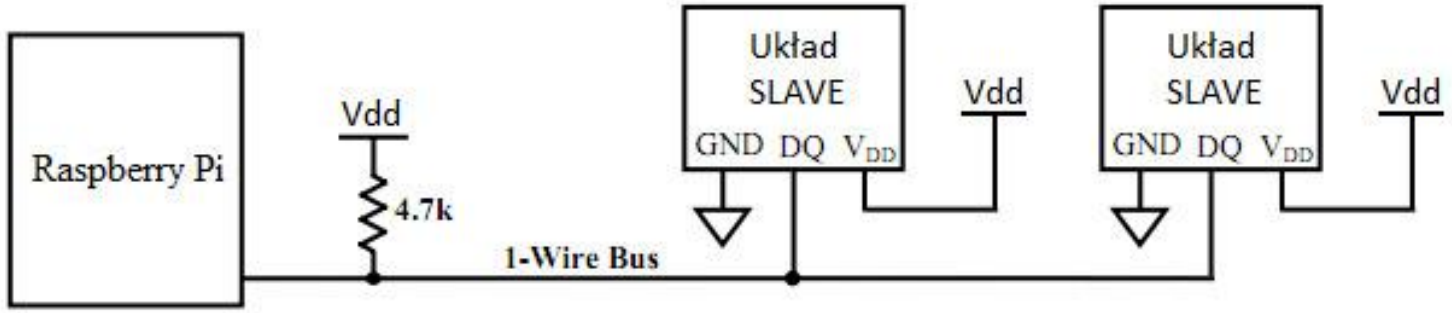
UART – > RS485 MODBUS ACSI/MODBUS RTU

- dwa przewody – prosta budowa, niski koszt
- interfejs prądowy, pętla prądowa
- bardzo wysoka odporność na zakłócenia E-M – transmisja różnicowa
- stosowany do budowy sieci przemysłowych
- długość do 1,2km
- jeden z najpopularniejszych interfejsów komunikacyjnych w przemyśle, automatyce domowe itd.
- wiele urządzeń pomiarowych dostępnych na rynku
- niska prędkość transmisji

Interfejsy komunikacyjne - przewodowe

1-wire

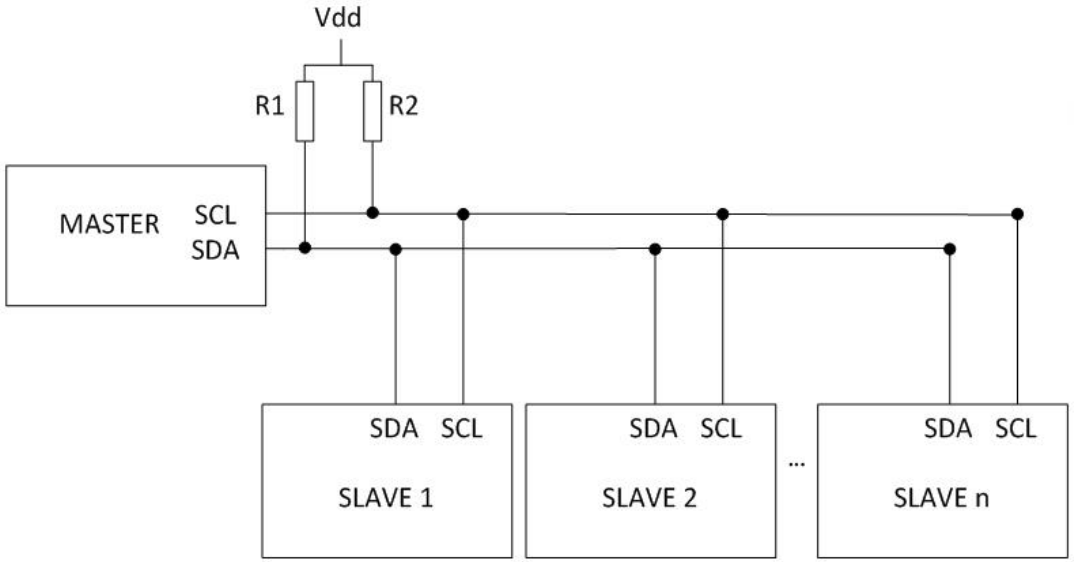
- 1 przewód
- wiele urządzeń, master-slave
- maksymalna odległość do 300m
- slave posiada stały fabryczny numer: 64-bitowy kod identyfikacyjny
- najczęściej wykorzystywana do prostych urządzeń pomiarowych

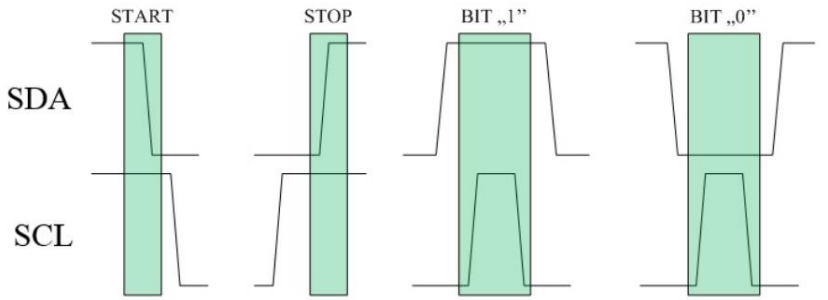


Interfejsy komunikacyjne - przewodowe

I2C

- 2 przewody, SCL linia zegarowa, SDA – linia danych
- wiele urządzeń, master-slave
- duża szybkość, zegar trybu szybkiego 400kHz - do 3,4 Mb/s
- slave posiada unikatowy adres, możliwość konfiguracji
- najczęściej wykorzystywana do czujników w obrębie płytki PCB – małe odległości

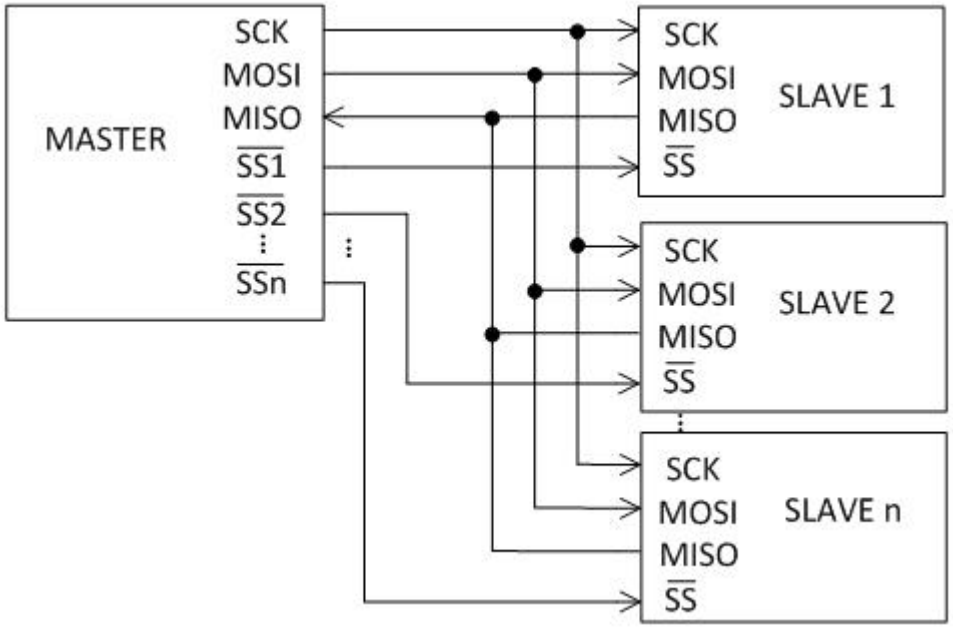




Interfejsy komunikacyjne - przewodowe

SPI

- 3 przewody + 1 select – SCK, MOSI,MISO , CS
- wiele urządzeń, master-slave
- bardzo duża szybkość przesyłania danych do 10 Mb/s
- brak zapisanych adresów, aktywny slave wybierany jest pinem
- najczęściej wykorzystywana do czujników w obrębie płytki PCB – małe odległości



Interfejsy komunikacyjne

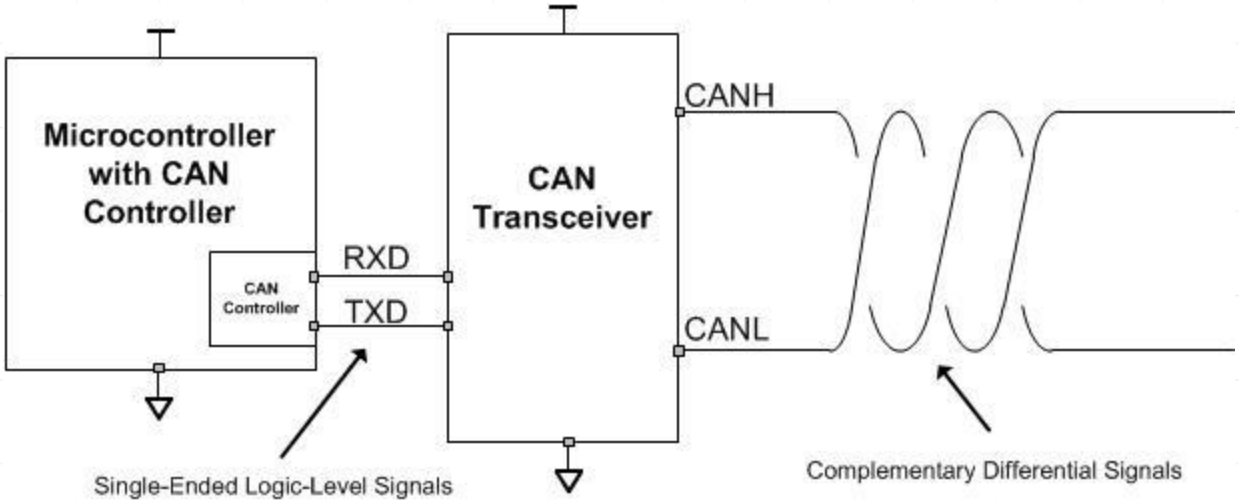
CAN – warstwa sprzętowa

Interfejs CAN (Controller Area Network) to szeregowa magistrala komunikacyjna powstała w latach 80. XX w. w firmie Robert Bosch GmbH z myślą o zastosowaniach w przemyśle samochodowym (ABS, sterowanie silnika).

Istotną cechą interfejsu CAN jest to, że wszystkie funkcjonalności warstwy fizycznej i łącza danych zostały zaimplementowane w układach scalonych realizujących transmisję CAN.

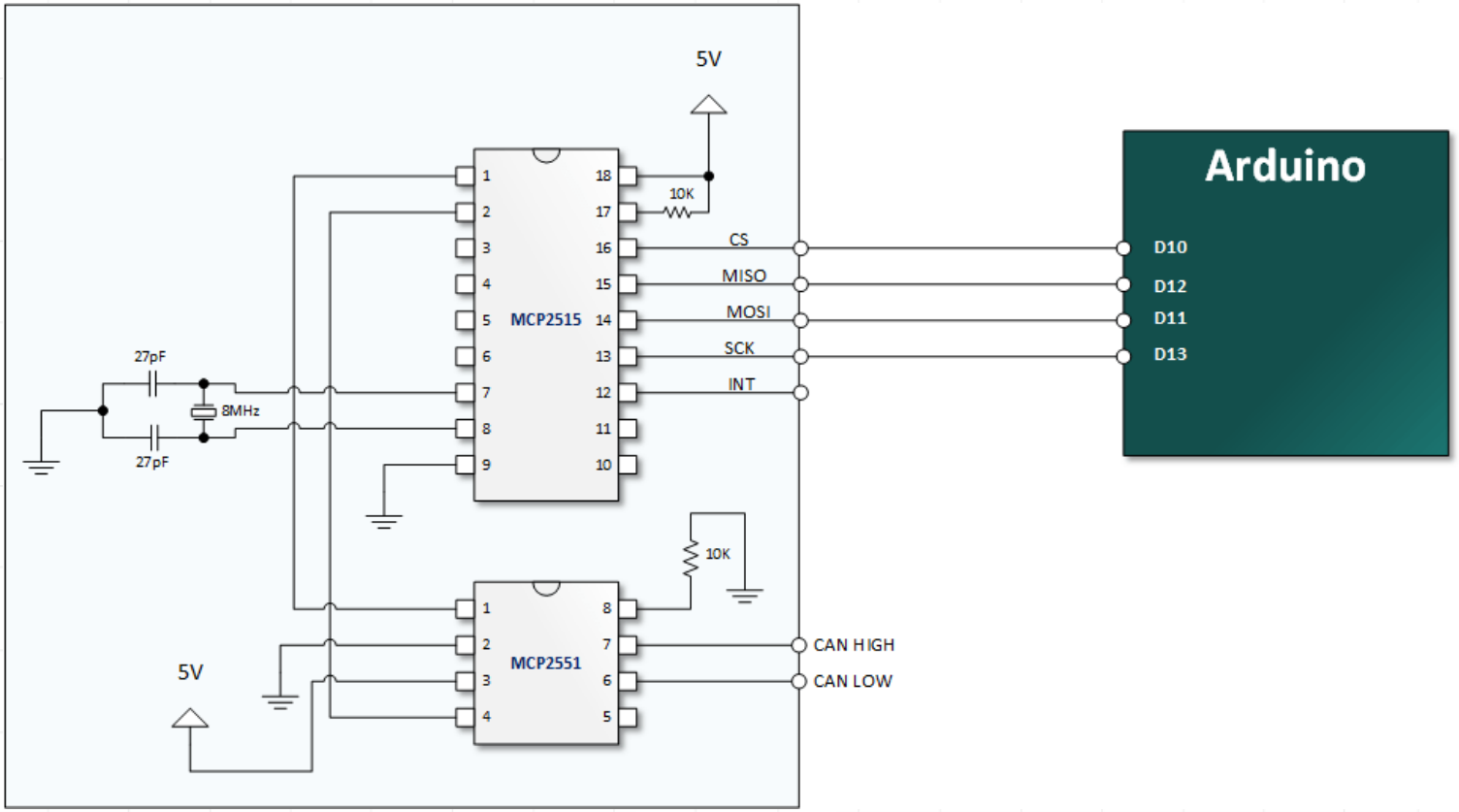
Interfejsy komunikacyjne

CAN – warstwa sprzętowa



Interfejsy komunikacyjne

CAN – warstwa sprzętowa



Interfejsy komunikacyjne

CAN

- magistrala szeregowa asynchroniczna
- magistrala komunikacyjna multi-master
- każdy odbiera wszystkie informacje występujące na magistrali
- filtry wiadomości wbudowane w kontroler CAN – odciążenie procesora
- priorytety wiadomości
- każdy może nadawać kiedy chce
- arbitraż rozwiązywany przez kontroler interfejsu CAN
- kontroler obsługuje błędy

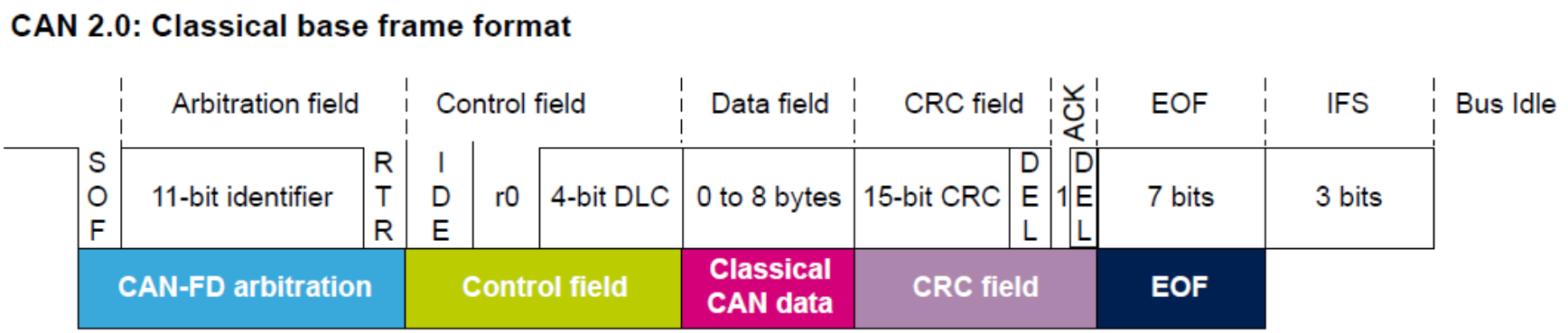
Interfejsy komunikacyjne

CAN – długość magistrali

Prędkość	Odległość
1 Mbit/Sec	40 metrów
500 kbits/Sec	100 metrów
100 kbits/Sec	500 metrów
50 kbits/Sec	1000 metrów

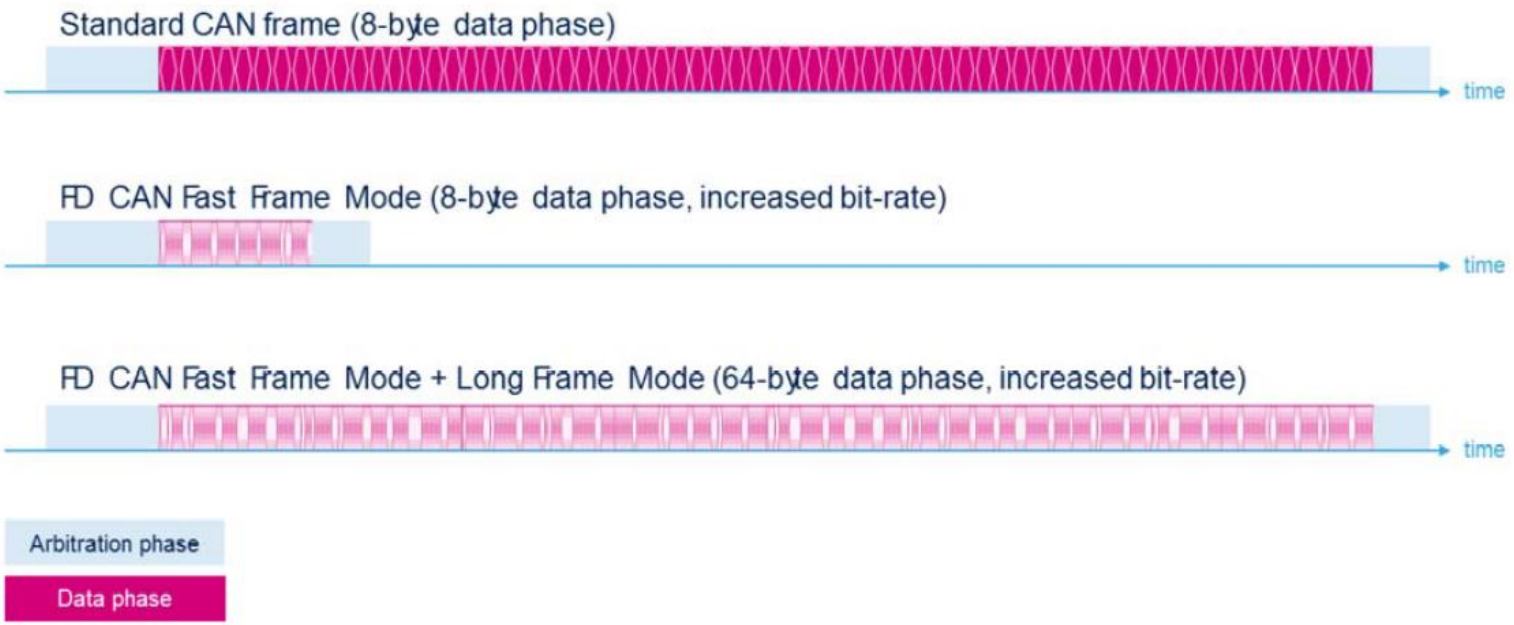
Interfejsy komunikacyjne

CAN – ramka



Interfejsy komunikacyjne

CAN – ramka



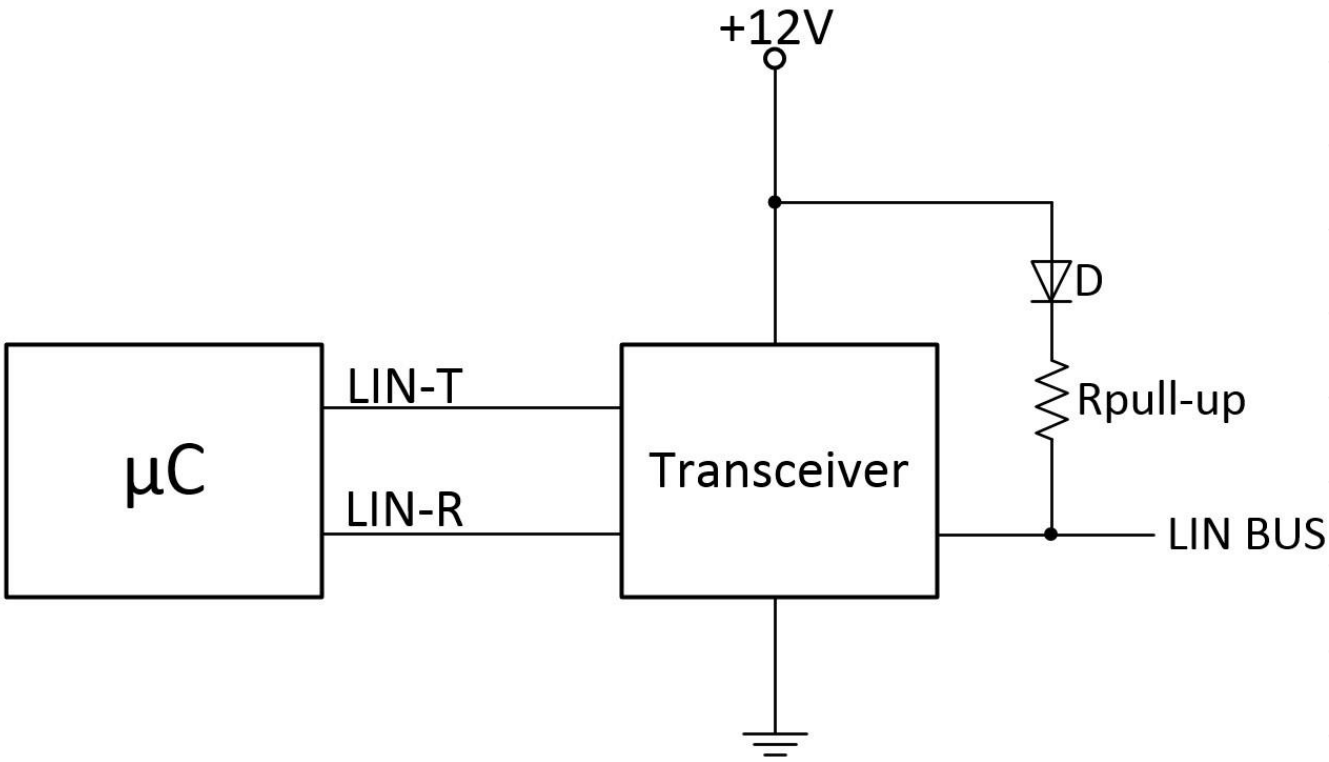
Interfejsy komunikacyjne

LIN

Protokół LIN stosowany jest w miejscach gdzie prędkość, oraz bezawaryjny przesył danych nie jest parametrem krytycznym.
LIN nie jest tak pewny/odporny jak CAN, lecz jest dużo tańszy, dlatego jest często używaną alternatywą.
Przykładami miejsc zastosowań protokołu LIN w samochodzie są: sterowniki szyb elektrycznych, sterowniki wycieraczek, czy lusterek.

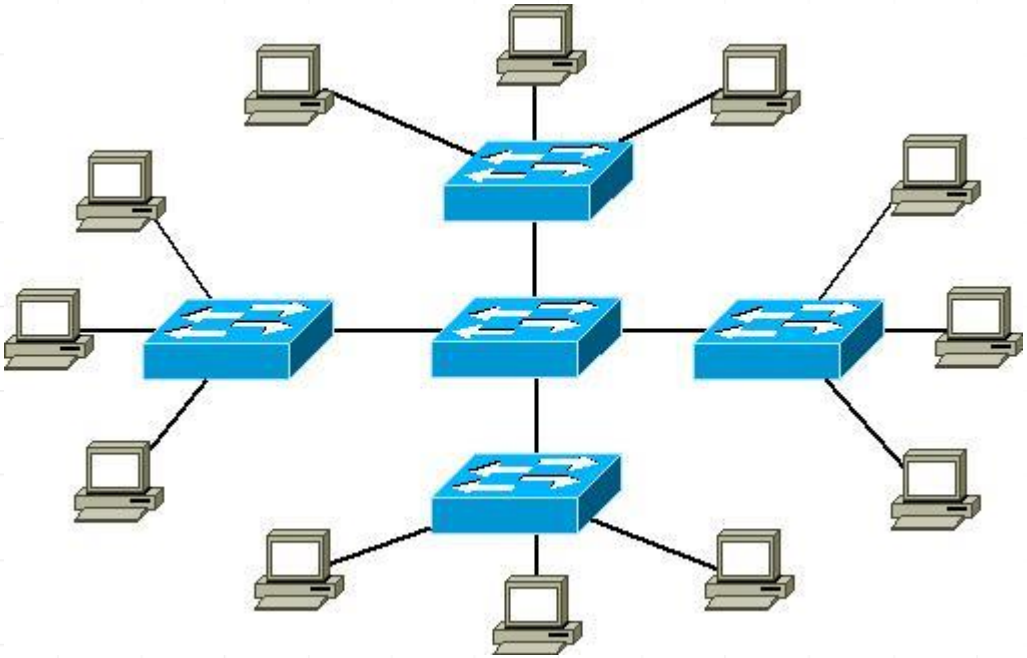
Interfejsy komunikacyjne

LIN – warstwa sprzętowa



Interfejsy komunikacyjne

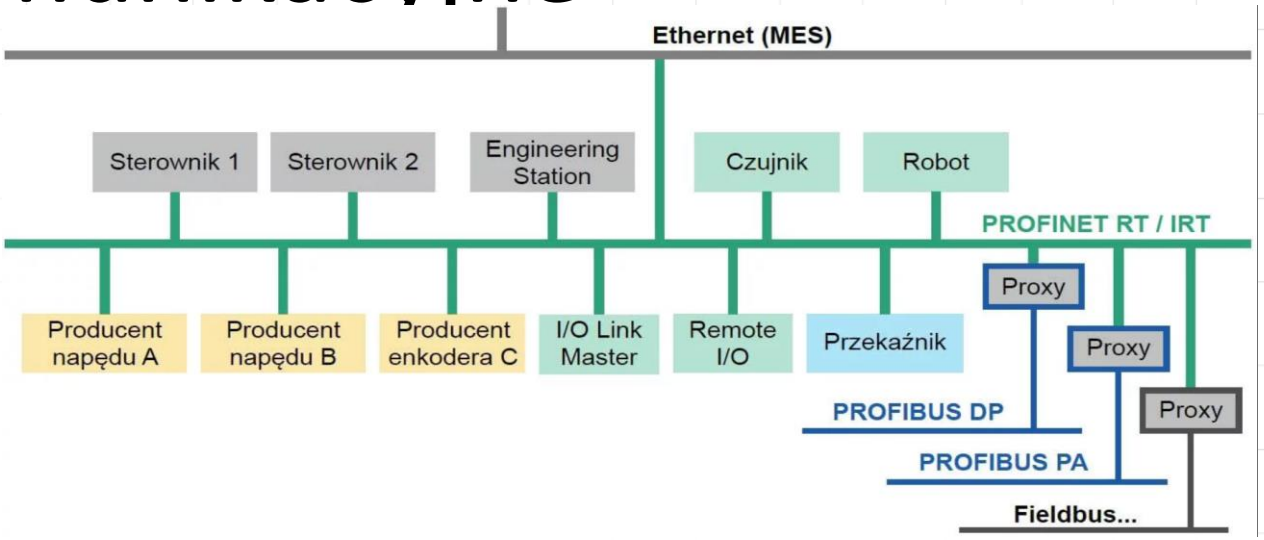
Ethernet

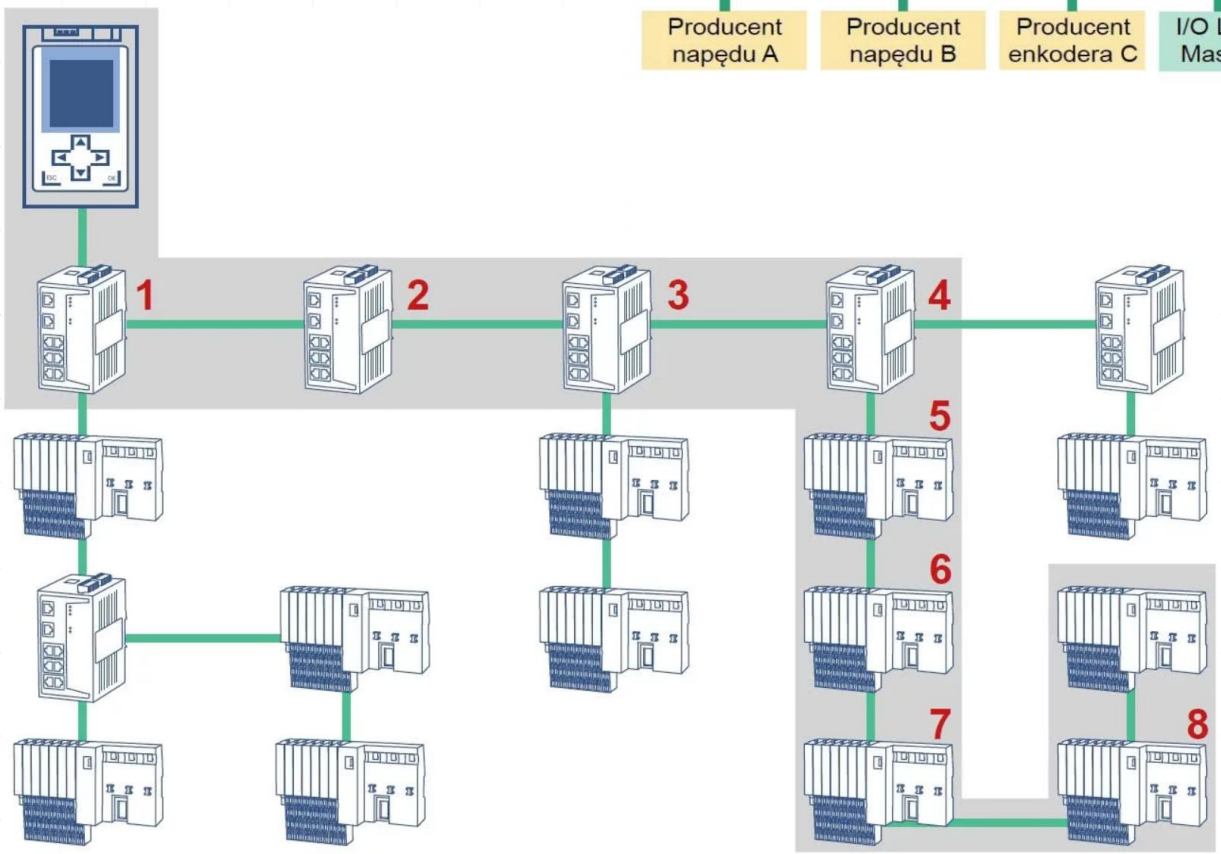


Wymaga stos komunikacyjny
obsługa standardowych
ramek danych

Interfejsy komunikacvine

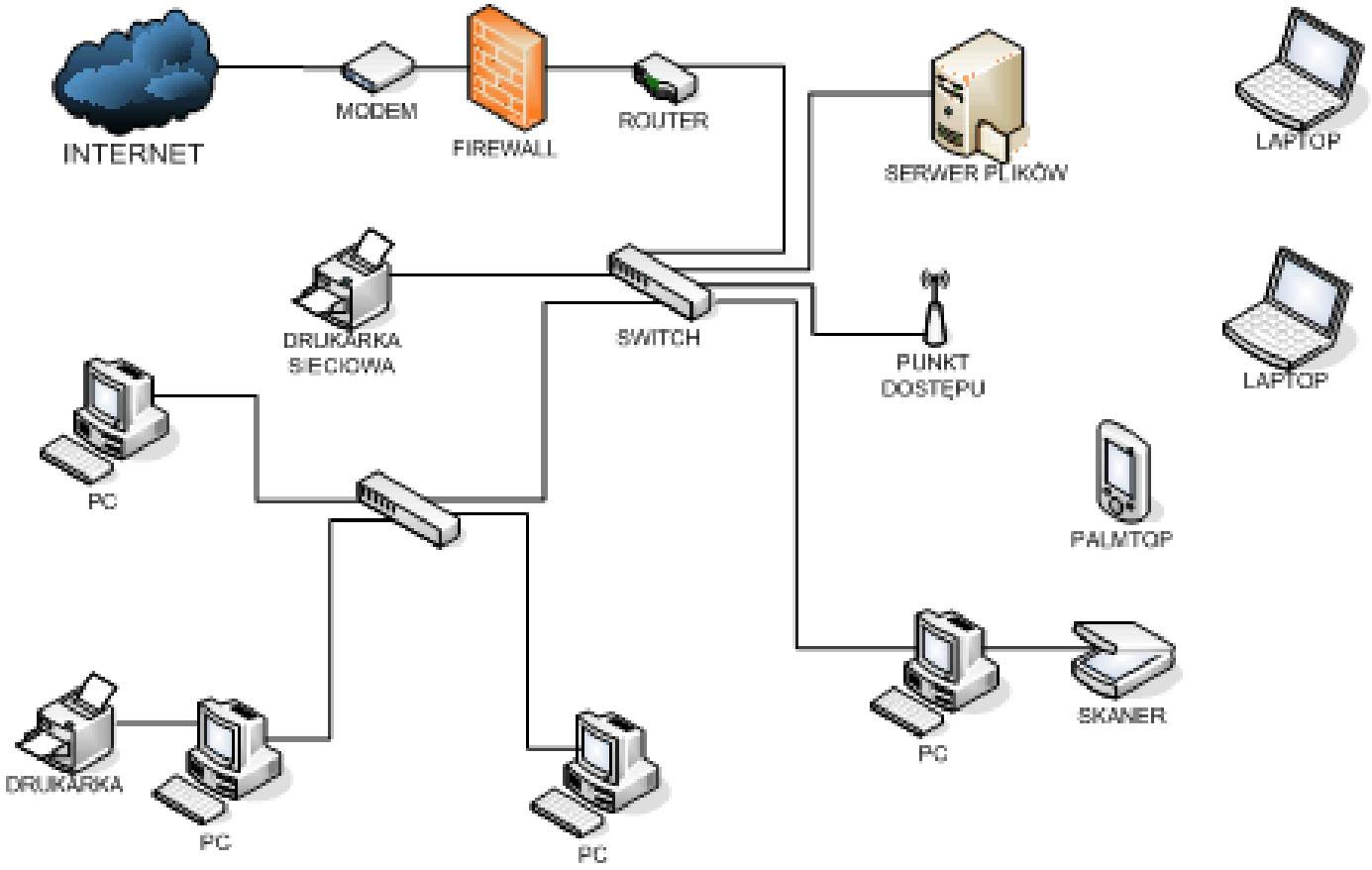
Ethernet przemysłowy profinet





Interfejsy komunikacyjne

Ethernet – Internet domowy



Interfejsy komunikacyjne

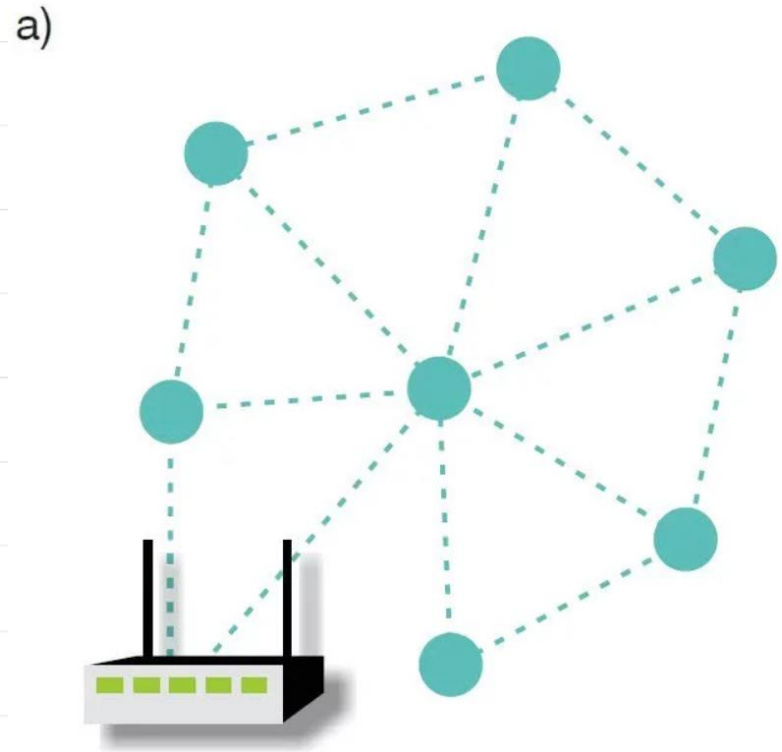
Komunikacja bezprzewodowa – porównanie z przewodową

Zalety:
- brak przewodu łączącego urządzenia, mobilność użytkowania
- brak problemu dopasowania napięciowego
- łatwe dodawania kolejnych urządzeń

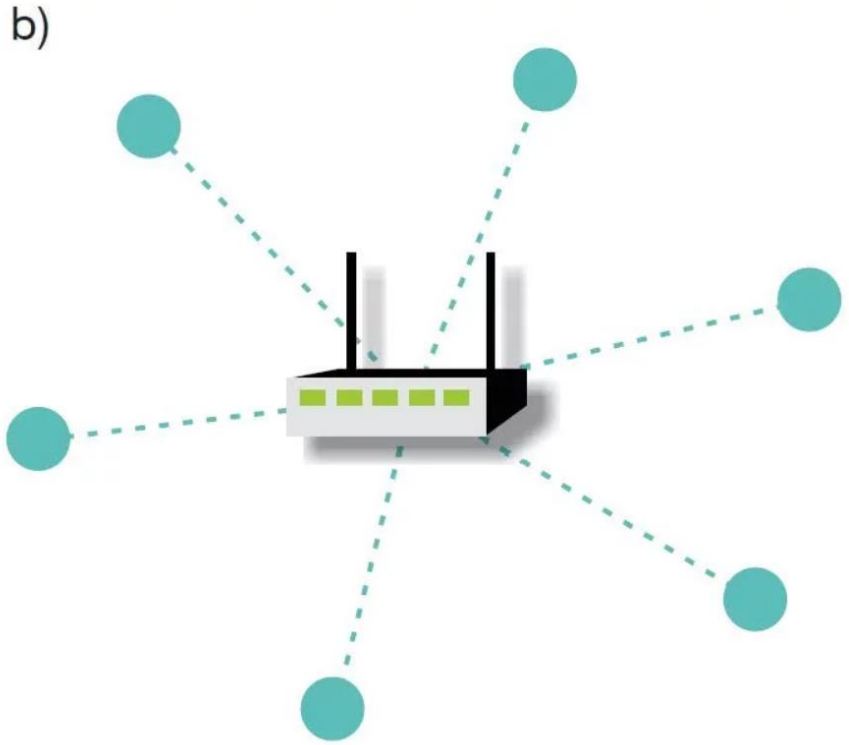
Wady:
- większa podatność na zakłócenia
- większe prawdopodobieństwo błędów transmisji
- większe opóźnienia
- łatwiejsza możliwość podsłuchiwania komunikacji
- prędkość zależna od odległości od nadajnika

Interfejsy komunikacyjne

Komunikacja bezprzewodowa – topologia



siatka - mesh



gwiazda

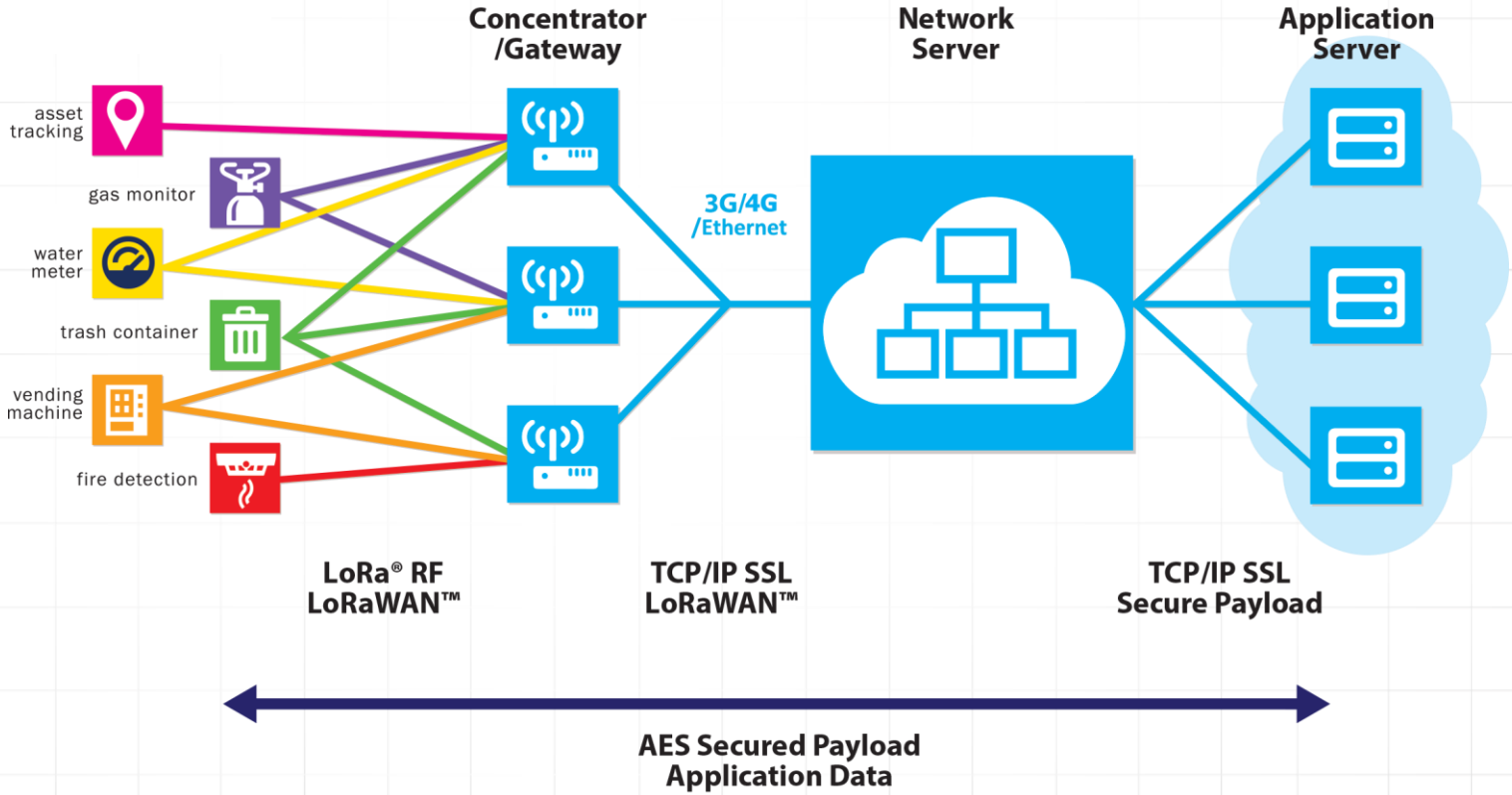
Interfejsy komunikacyjne

Komunikacja bezprzewodowa – 433MHz, 868MHZ

- otwarte i ogólnodostępne pasmo
- komunikacja lokalna
- często używana:
  - piloty do bram
  - zdalne sterowanie urządzeń
  - stacje pogodowe
  - IoT

Interfejsy komunikacyjne

Komunikacja bezprzewodowa – LoRa – 868MHz



Interfejsy komunikacyjne

Komunikacja bezprzewodowa – LoRa – 868MHz

Zalety:

- bardzo duży zasięg do 30km (2-5km w mieście)
- energooszczędna – praca czujnika na baterii do 5-10lat
- szybkość instalacji

Wady:

- niska prędkość wysyłania danych
- stworzona do okresowego wysyłania danych pomiarowych

Interfejsy komunikacyjne

Komunikacja bezprzewodowa – wifi

Zalety:

- brak przewodu łączącego urządzenia – wygoda użytkowania
- brak problemu dopasowania napięciowego
- duża odporność na wyładowania atmosferyczne
- szybkość instalacji

Wady:

- większa podatność na zakłócenia – pasmo 2.4GHz bardzo popularne i wykorzystywane przez wiele urządzeń w domu
- łatwiejsza możliwość podsłuchiwania komunikacji
- mały zasięg
- prędkość zależna od odległości od nadajnika

Interfejsy komunikacyjne

Komunikacja bezprzewodowa – bluetooth

Zalety:

- energooszczędna – dla urządzeń IoT
- zasięg do 400m – można regulować w zależności od prędkości transmisji
- przewidziana do komunikacji małego zasięgu
- Bluetooth Mesh Networking - nowe standardy implementują rozszerzenia
- lokalizacja urządzeń

Wady:

- niższa przepustowość w porównania do WiFi
- wymagana implementacja skomplikowanego protokołu komunikacyjnego

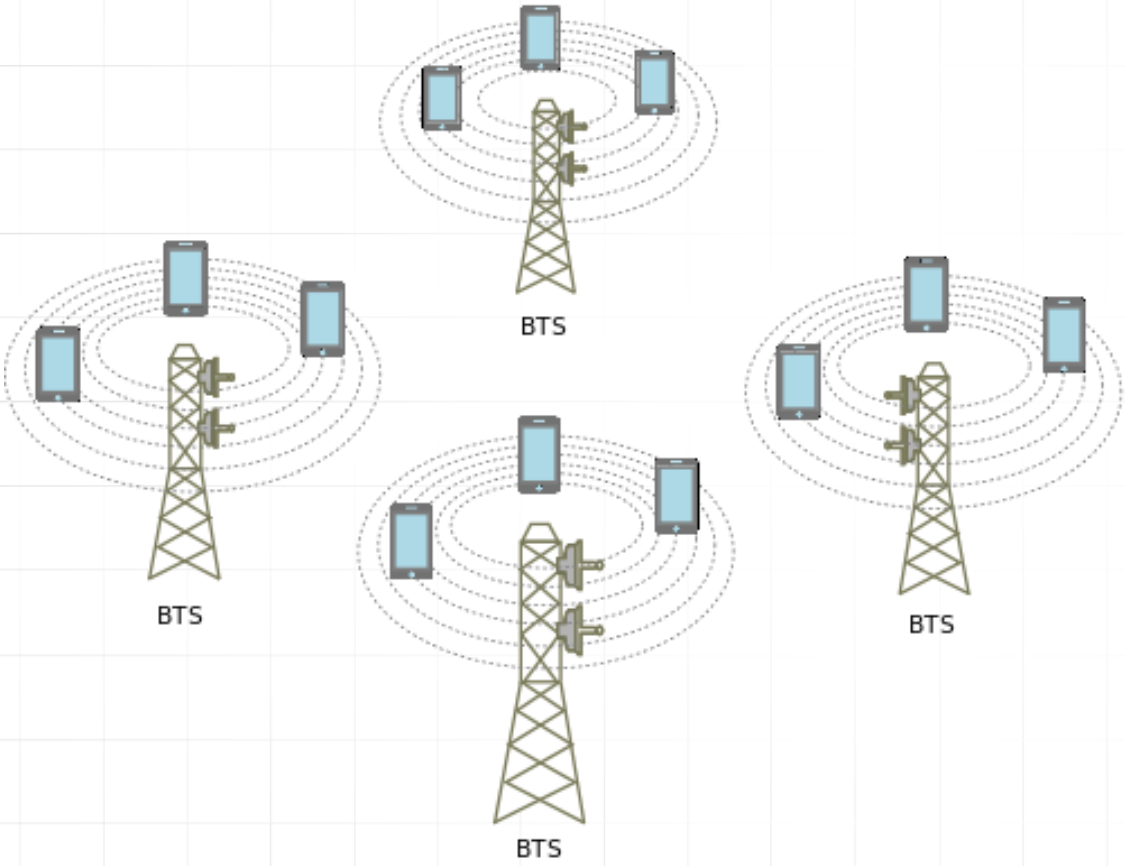
Interfejsy komunikacyjne

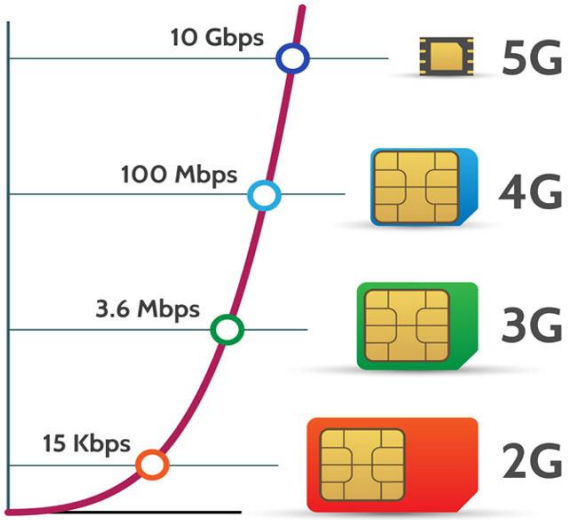
Komunikacja bezprzewodowa – bluetooth

	Wi-Fi	BLE	LoRaWAN
Maksymalna szybkość transmisji	72 Mbps	2 Mbps	50 kbps
Zasięg	100 m	400 m	15 km
Topologia	Gwiazda	Punkt-punkt/Siatka	Gwiazda
Obsługa protokołu IP przez węzły końcowe	Tak	Nie	Nie
Wsparcie przez PC oraz urządzenia mobilne	Tak	Tak	Nie
Dostępność gotowej infrastruktury	Tak (AP)	Tak (urządzenia mobilne)	Lokalni operatorzy na niektórych obszarach

Interfejsy komunikacyjne

Komunikacja bezprzewodowa – GSM





Interfejsy komunikacyjne

Komunikacja bezprzewodowa – GSM

Zalety:

- brak potrzeby budowania własnej infrastruktury
- nieograniczony zasięg (wiele BMS)
- ciągle rozwijana i utrzymywana przez zewnętrzną firmę
- szybka reakcja serwisowa w razie awarii
- niski koszt początkowy

Wady:

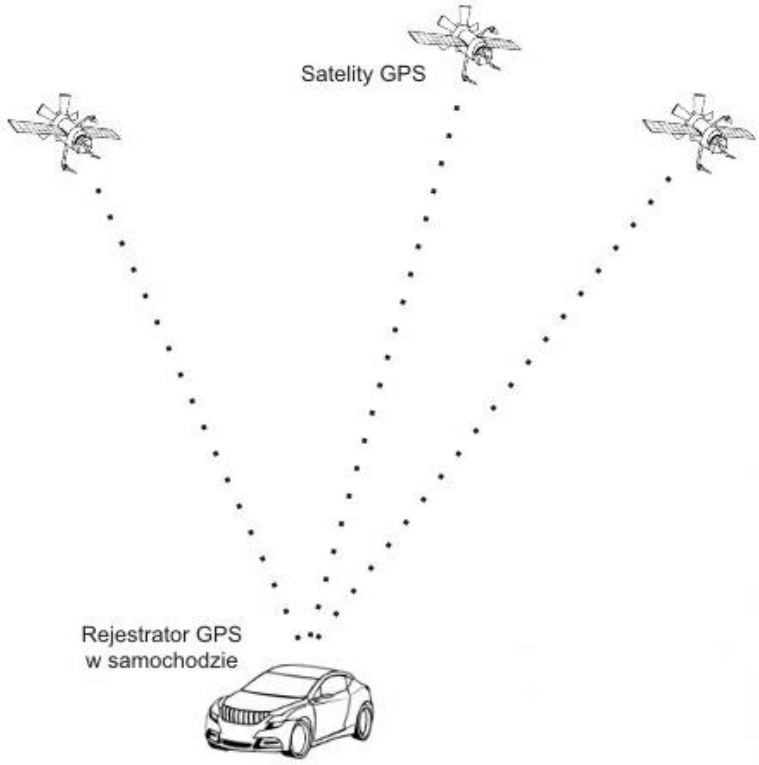
- zależność od operatora zewnętrznego
- opłata abonamentowa i za transfer
- koszt rośnie przy większej ilości urządzeń

Interfejsy komunikacyjne

Komunikacja bezprzewodowa – GPS

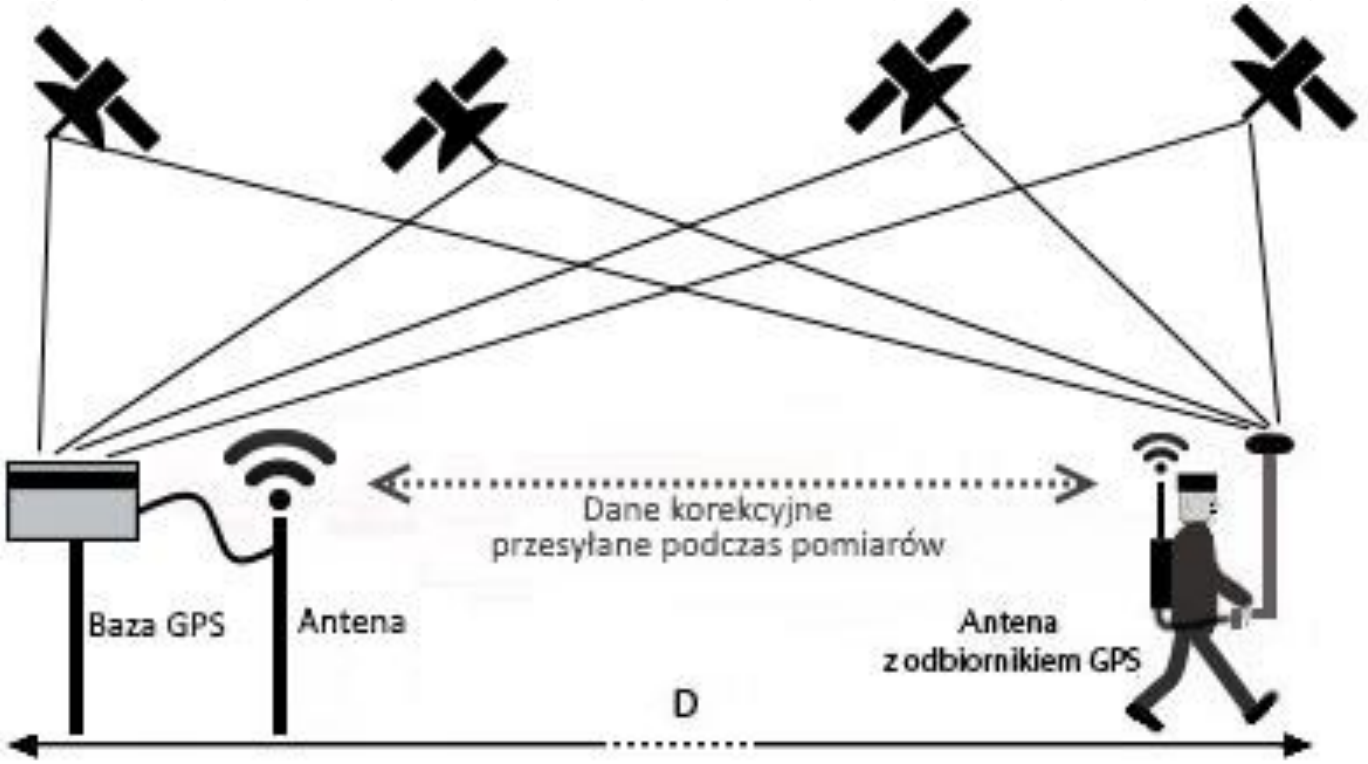
- pozycjonowanie obiektów
- różne systemy:
  - GPS
  - Glonass
  - Galileo
  - BeiDou

- dokładność pozycjonowania 1-3m
- GPS RTK – dokładność do 1-2 cm



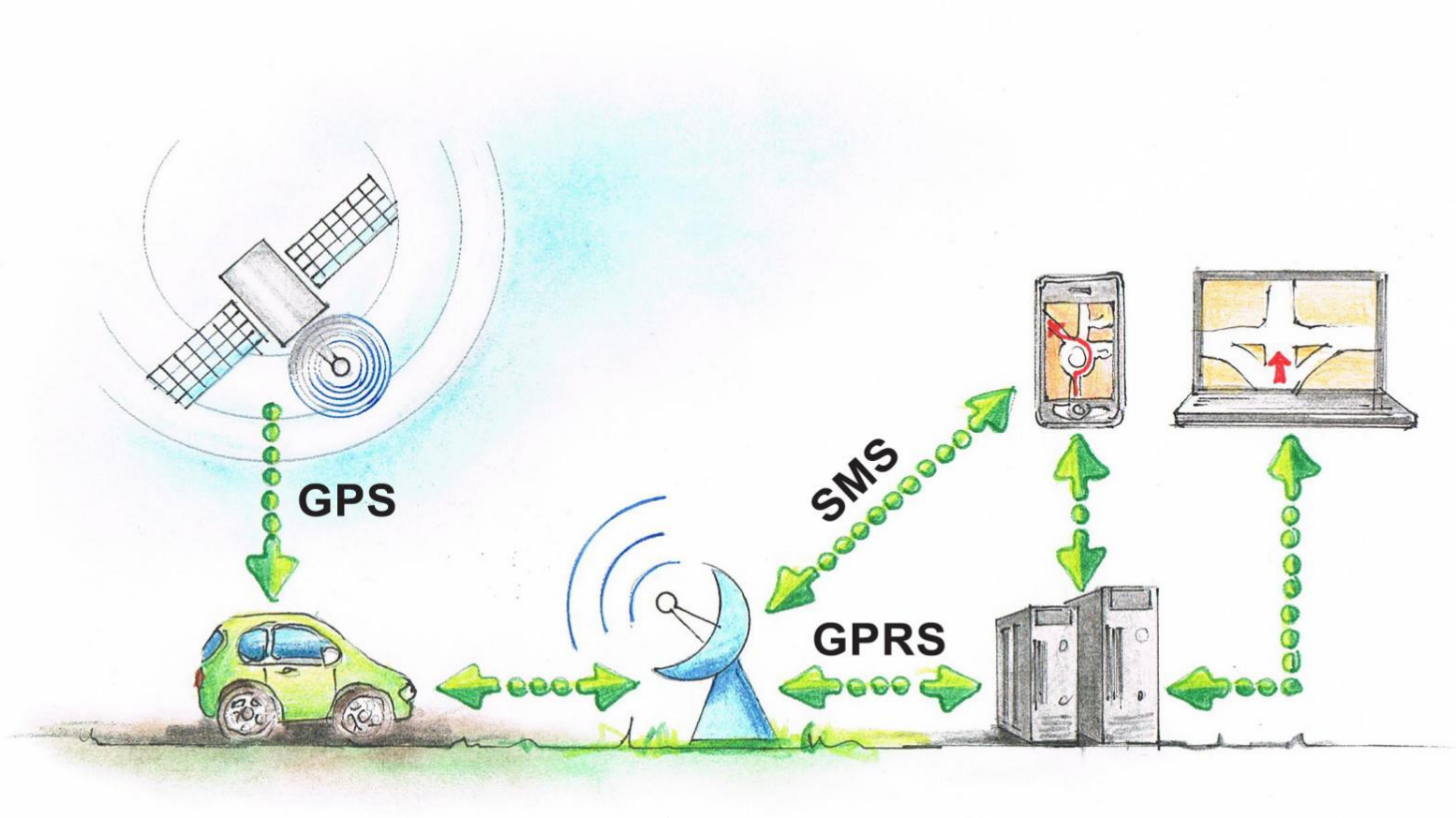
Interfejsy komunikacyjne

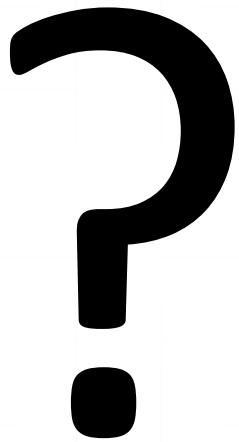
Komunikacja bezprzewodowa – GPS RTK



Interfejsy komunikacyjne

Komunikacja bezprzewodowa – GPS – GSM - lokalizacja





Interfejsy komunikacyjne

Komunikacja przewodowa – komunikacja bezprzewodowa
RS485 – WiFi

???
Co lepsze???