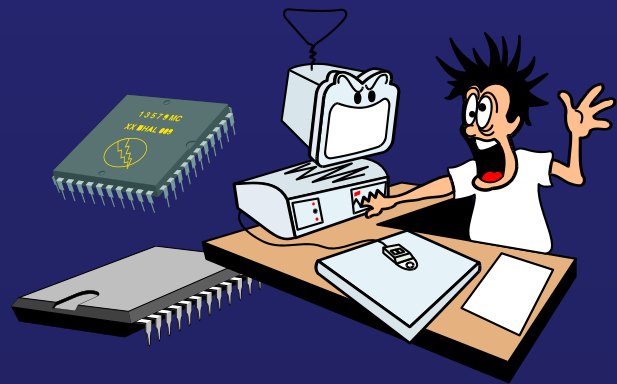
WYDZIAŁ FIZYKI
i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Systemy wbudowane

Witold Kozłowski



https://std2.phys.uni.lodz.pl/mikroprocesory/

Systemy wbudowane

Kierunek: Informatyka
PRACOWNIA DYDAKTYCZNA

Uwaga !!!

Proszę o wyłączenie telefonów komórkowych

na wykładzie i laboratorium

Systemy wbudowane

Kierunek: Informatyka
PRACOWNIA DYDAKTYCZNA

Wykład 9.

Interfejs szeregowy RS232

Interfejs

Bardzo często w praktyce trzeba połączyć komputer z jakimś urządzeniem pomiarowym, np.:

woltomierz, amperomierz, czujnik temperatury, zasilacz wysokiego napięcia, wagę elektroniczną, kasę fiskalną itp..

Z reguły wymagane przez to urządzenie poziomy prądów i napięć są różne od dostarczanych przez PC. Również szybkość przestrajania się i działania takich przyrządów bywa różna i nie zawsze jest zgodna z szybkością działania PC. Z tego względu niezbędnym jest stosowanie specjalnych układów, zapewniających odpowiednie wzajemne dopasowanie urządzeń i komputera, tzw. interfejsów

Interfejs

W tłumaczeniu z angielskiego *interface* oznacza obszar wzajemnego oddziaływania (niekiedy wyraz ten tłumaczy się dosłownie jako „sprzęg”)

Polską normą PN-83/T-0653 określa ściśle definicje systemu interfejsu. Z jej treści możemy wywnioskować, że system interfejsu jest to zbiór niezależnych od urządzeń elementów mechanicznych, elektrycznych i funkcjonalnych, koniecznych w procesie wymiany informacji pomiędzy urządzeniami.

Wymiana danych pomiędzy komputerem a urządzeniami (lub pomiędzy dwoma urządzeniami) realizowana jest dzięki wcześniejszemu ustaleniu protokołu transmisji, czyli specyficznego zbioru reguł, procedur lub różnego rodzaju konwencji dotyczących formatu i czasu trwania przesyłania danych

Interfejs

Transmisja szeregowa

Do najważniejszych standardów, realizujących transmisję szeregową należy zaliczyć: RS 232C (w Europie zwany V.24) i jego rozwinięcia:

- RS 422,

- RS 423,

- RS 449,

- RS 485,

- IEEE 1394 stosowany głównie w urządzeniach przetwarzających dźwięk i obraz

-(*Wi-Fi*),

- oraz najnowszy produkt USB (ang. *Universal Serial Bus*)

Interfejs RS232

Podstawową wersję RS 232 (ang. *Recommended Standard*) wprowadzono w 1962 roku w USA. Początkowo standard ten miał służyć jedynie do obsługi modemów.

Największą popularność zdobyła wersja RS 232C wprowadzona w 1969 roku, zaś oficjalnie do rangi standardu została podniesiona w roku 1986.

RS 232C jest powszechnie stosowanym i akceptowanym standardem dla szeregowej wymiany danych cyfrowych pomiędzy urządzeniem:

DTE (ang. *Data Terminal Equipment*), obecnie utożsamianym z komputerem

DCE (ang. *Data Communication Equipment*) — urządzeniem zewnętrznym (w oryginale modemem).

W sposób jednoznaczny definiuje on parametry elektryczne, mechaniczne i logiczne łącza szeregowego. Oficjalna jego nazwa brzmi:

Interface Between Data Terminal and Data Circuit Termination Equipment Employing Serial Binary Data Interchange

RS 232C stosowany bywa wszędzie tam, gdzie mniej istotną rolę odgrywa przepustowość łącza, natomiast ważna jest niezawodność i prostota obsługi protokołu komunikacyjnego.

Interfejs RS232

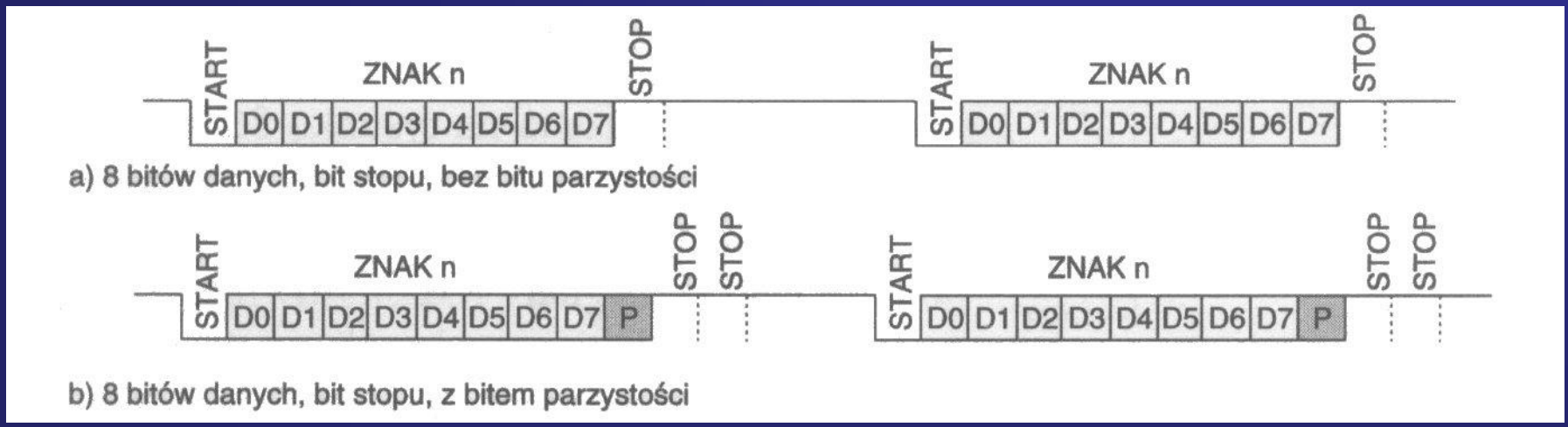
Komputery osobiste wyposażone są w łącza szeregowe przystosowane do transmisji asynchronicznej, tzn. komputer i urządzenie muszą pracować z jednakową, wcześniej uzgodnioną prędkością oraz taką samą strukturą znaków.

Transmisja taka może być realizowana w trybie bez potwierdzenia odbioru lub z potwierdzeniem odbioru. Drugi sposób zapewnia nam możliwość kontrolowania poprawności wysyłanych–odbieranych danych. Dane przesyłane są w postaci tzw. ramki (ang. *frame*), która jest najmniejszą porcją możliwej do przesłania informacji. Bity przesyłane są kolejno.

Do kodowania znaków stosuje się najczęściej kod ASCII (*American Standard Code of Information Interchange*).

Interfejs RS232

Obecnie ramka może zawierać od 5 do 8 bitów danych (jednak większość spotykanych urządzeń posługuje się słowem 7 lub 8 bitowym) poprzedzonych bitem startu oraz zakończonych bitem parzystości i jednym lub więcej bitami stopu. Przed rozpoczęciem transmisji bit startu przyjmuje zawsze wartość 0, zaznaczając wyraźnie moment początkowy. Odwrotność czasu trwania transmisji jednego bitu określa szybkość przesyłu w bitach na sekundę



Bit kontroli parzystości przesyłany za ostatnim bitem danych jest jedną z metod monitorowania poprawności transmitowanych danych. Z reguły przyjmuje dwie wartości: 0 lub 1. Ilości jedynek w polu danych może być uzupełniana do liczby parzystej (*evenparity*) lub nieparzystej (*oddparity*).

Interfejs RS232

RS 232C jest interfejsem cyfrowym, zatem jego poziomom logicznym (0–1) należy przypisać określone przedziały napięć zarówno ujemnych jak i dodatnich, które są z reguły nieco wyższe od stosowanych w komputerze. Pozwala to w dużym stopniu uniezależnić sygnał na wejściu interfejsu od przypadkowych zakłóceń.

Dla sygnałów sterujących:

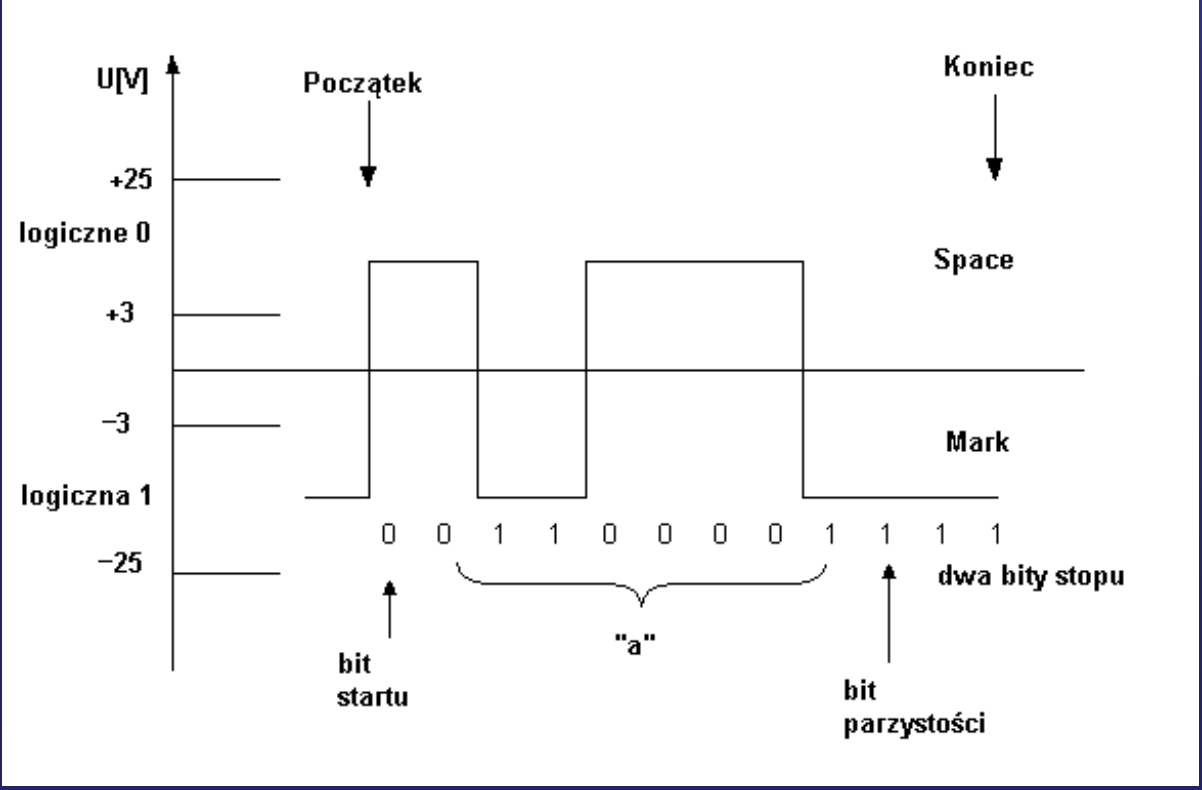
Logicznej 1 odpowiada przedział od -3 do -25V, tzw. stan aktywny, wysoki, włączony lub „ON”.

Logicznemu 0 odpowiada przedział od +3 do +25 V, jest to stan nieaktywny, niski, wyłączony lub „OFF”.

Interfejs RS232

Łącze w trakcie ciszy utrzymywane jest w stanie logicznej 1. Transmisja rozpoczyna się od bitu startu, który zawsze przyjmuje wartość logicznego 0. Po nim następuje transmisja ośmiu bitów reprezentujących znak. Później jest bit parzystości, potem dwa bity stopu zamykające ramkę. Po bicie stopu łącze wraca do stanu ciszy.

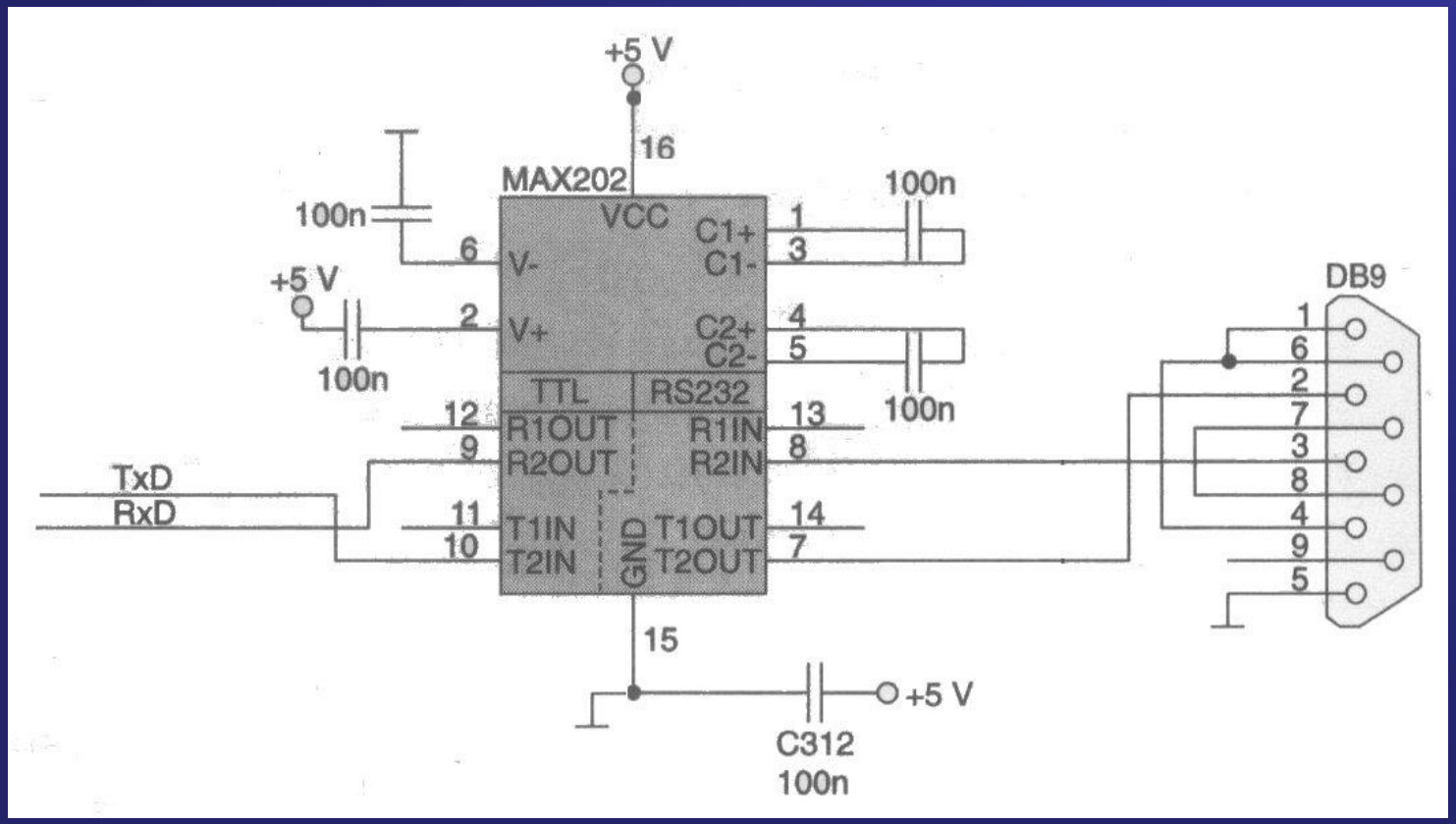
Jeden lub dwa bity stopu stosowane są po to, by odbiornik i nadajnik mogły dokonać wzajemnej synchronizacji przed transmisją kolejnej ramki danych.



Na rysunku pokazano przebieg czasowy przykładowej ramki, symbolizującej wysłanie jednej litery ”a” reprezentowanej na ośmiu bitach — dziesiętnie 97, binarnie 01100001. Bit parzystości został ustawiony jako *markparity*, zastosowano też dwa bity stopu.

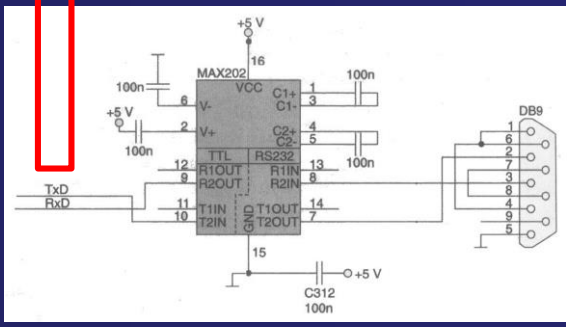
Interfejs RS232

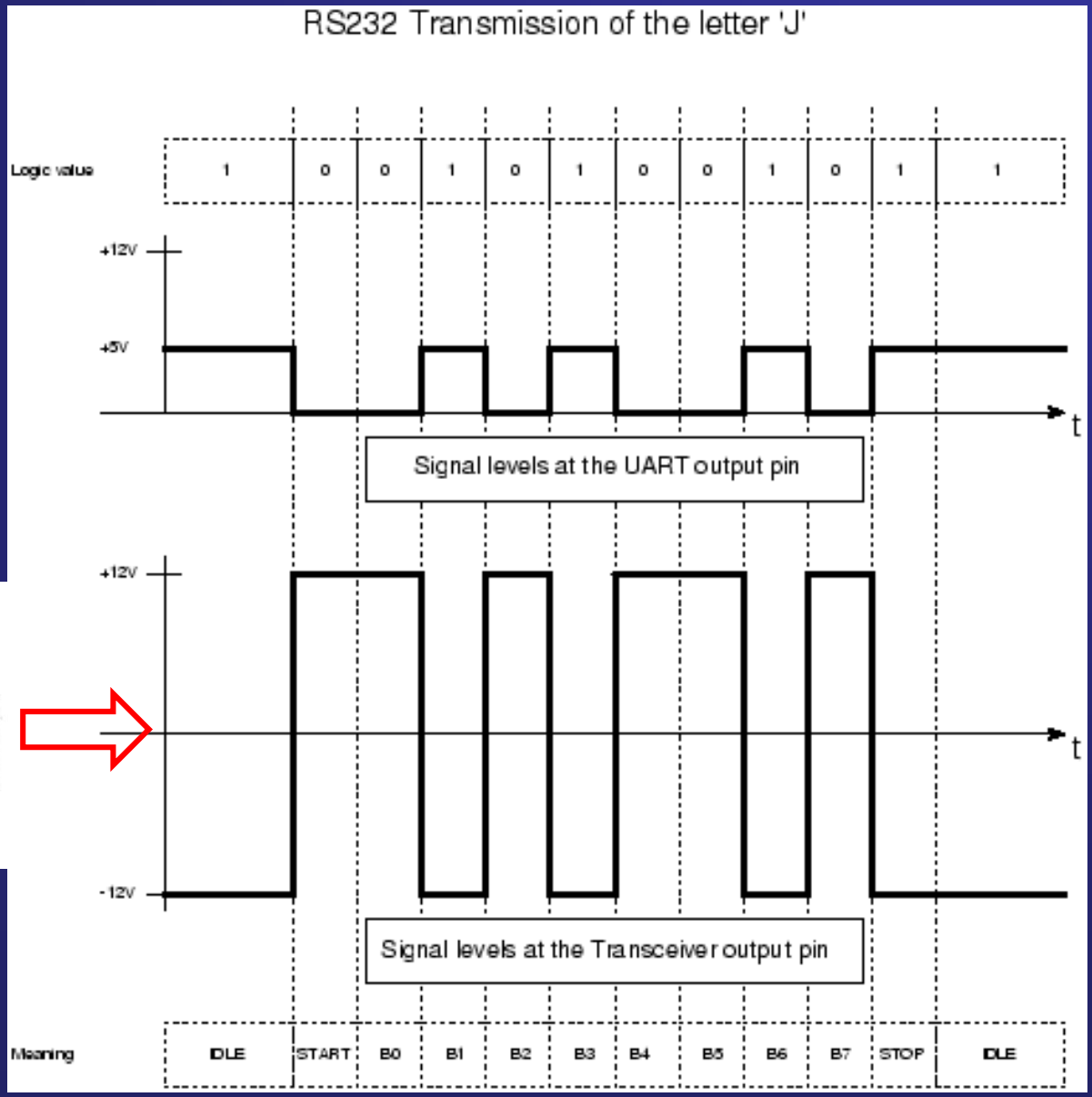
W praktyce układy nadajników np. (MAX202) zasilane są napięciem ±12V, dając amplitudę sygnałów ±8V. W tej sytuacji bitom 0 oraz 1 transmitowanego bajtu odpowiadają napięcia odpowiednio +12V oraz –12V[1]



[1] **Zgodnie z zaleceniami protokołu V.28 CCITT (Międzynarodowy Komitet Doradczy ds. Telefonii i Telegrafii) logicznemu zeru powinien odpowiadać potencjał dodatni +3V...+15V, zaś logicznej jedynce potencjał ujemny –3V... – 15V.**

Interfejs RS232

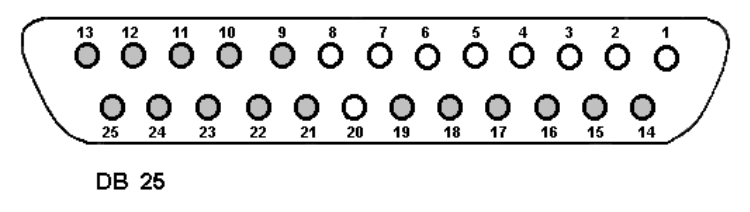


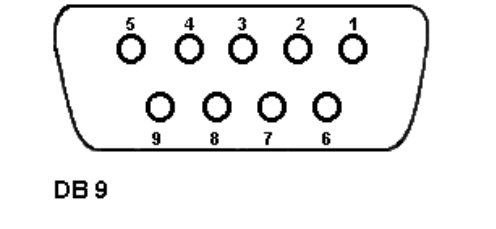


Interfejs RS232

Standardową linię interfejsu RS 232C stanowi 25-żyłowy przewód, przy czym większość z tych linii wykorzystuje się dla potrzeb transmisji synchronicznej.

W standardzie IBM wykorzystuje się jedynie 9 sygnałów, które są wystarczające do zrealizowania transmisji asynchronicznej. W komputerach PC używano początkowo dwóch rodzajów złączy szeregowych: 9- oraz 25-końcówkowych typu DB-9 i odpowiednio DB-25. W komputerach zaopatrzonych w nowsze płyty główne spotyka się jedynie złącza DB-9.

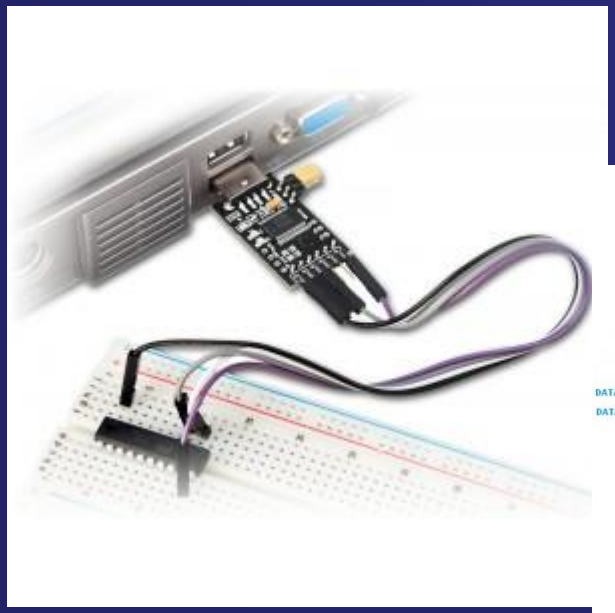


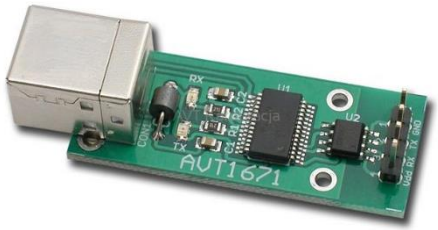


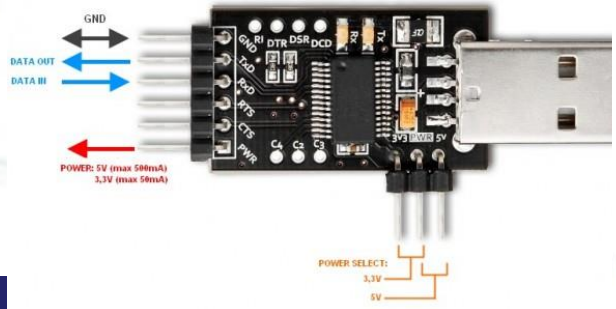
Konwertery USB UART

Konwerter umożliwia komunikacje pomiędzy dwoma popularnymi interfejsami szeregowymi USB i UART. Moduł oparty na niezawodnym układzie FT232RL firmy FTDI, zapewnia stabilną pracę oraz współpracę z najpopularniejszymi systemami operacyjnymi.

Moduł może być wykorzystany do wymiany danych między komputerem a systemem mikrokontrolerowym lub do programowania zestawów elektronicznych z wyprowadzonym interfejsem







Interfejs RS232

Linie TxD oraz RxD są przeznaczone do obustronnego przesyłania danych. Nazywamy je liniami danych. Pozostałe zaś są liniami sterującymi lub kontrolnymi (oczywiście za wyjątkiem linii masy). Ogólnie sygnały przekazywane łączem RS 232C można podzielić na trzy grupy:

1) sygnały danych: RxD, TxD,

2) sygnały sterujące urządzeniem zewnętrznym: RTS, DTR,

3) sygnały odbierane od urządzenia (kontrolne): CTS, DSR, RI, RLSD (DCD).

DB-25	DB-9	Opis sygnału	Kierunek sygnału
1	—	PG *Protective Ground* — masa ochronna	—
2	3	TxD *Transmitted Data* — dane wysyłane	Wyjście DTE (PC)
3	2	RxD *Received Data* — dane odbierane	Wejście DTE
4	7	RTS *Request To Send* — żądanie nadawania. PC zgłasza do urządzenia gotowość odbioru danych	Wyjście DTE
5	8	CTS *Clear To Send* — gotowość do wysyłania danych. Urządzenie potwierdza przyjęcie sygnału RTS	Wejście DTE
6	6	DSR *Data Set Ready* — odbiornik gotowy do odbioru danych wysłanych przez komputer	Wejście DTE
7	5	SG *Signal Ground* — masa sygnałowa	—
8	1	(RLSD) DCD *Data Carier Detect* — odbiór fali nośnej. Linia wykorzystywana głównie przez modemy	Wejście DTE
20	4	DTR *Data Terminal Ready* — gotowość komputera do odbierania-wysyłania danych.	Wyjście DTE
22	9	RI *Ring Indicator* — wskaźnik wywołania. Linia wykorzystywana głównie przez modemy	Wejście DTE

Interfejs RS232

Interfejs RS232 występuje w mikrokontrolerach w postaci wbudowanego układu pod nazwą UART (*Uniwersal Asynchronus serial Resiver and Transmitter*). Jest to uniwersalny asynchroniczny układ nadawczo odbiorczy do transmisji szeregowej. W większości współczesnych mikrokontrolerów układ ten może być także konfigurowany do synchronicznej transmisji danych i wówczas jest określany akronimem USART (*Uniwersal Synchronus and Asynchronus serial Resiver and Transmitter*).

Bascom nie ma możliwości zadeklarowania trybu asynchronicznego albo synchronicznego. Możliwa jest praca USART jak UART, tzn. w trybie asynchronicznym.

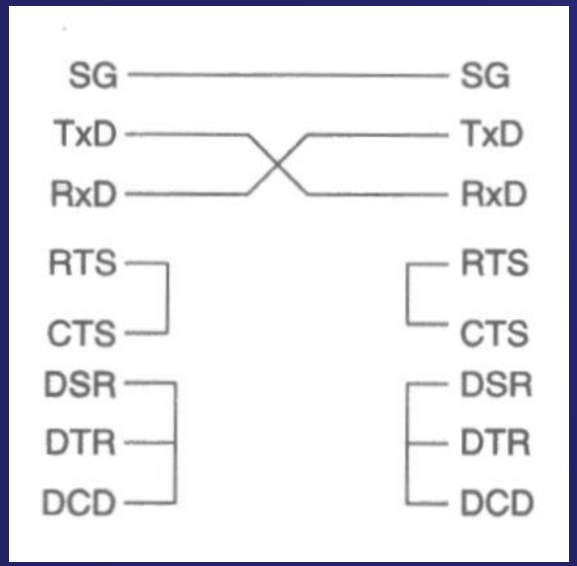
Do przesyłania danych w interfejsie RS232 wykorzystywane są dwie linie:

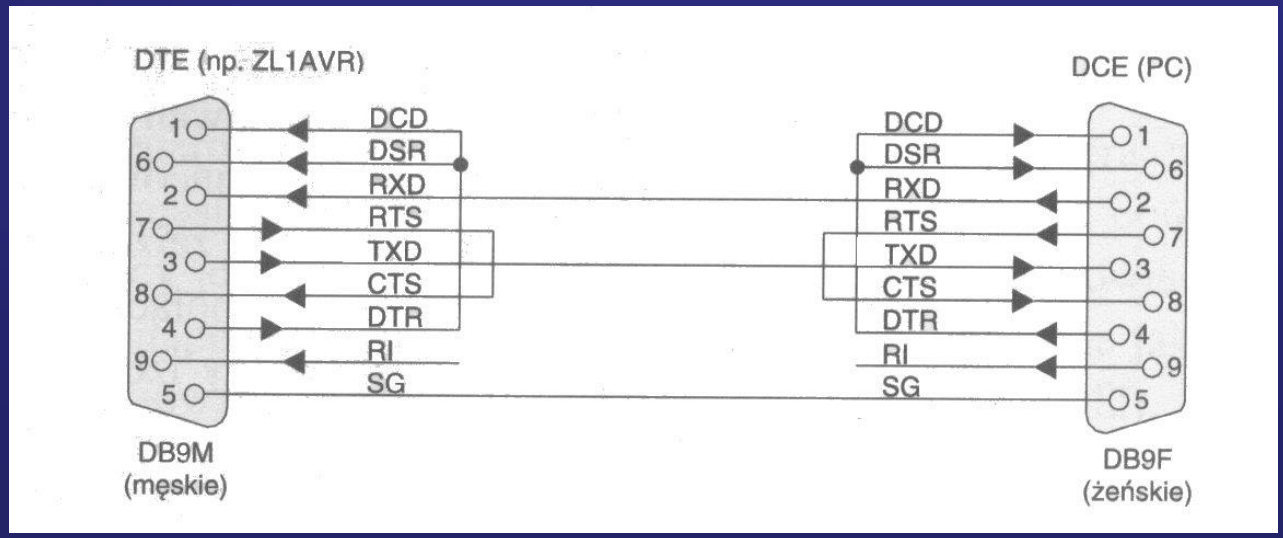
- **TxD – linia, którą są wysyłane dane z mikrokontrolera ,**
- **RxD – linia, którą są odbierane dane przez mikrokontroler.**

Interfejs RS232

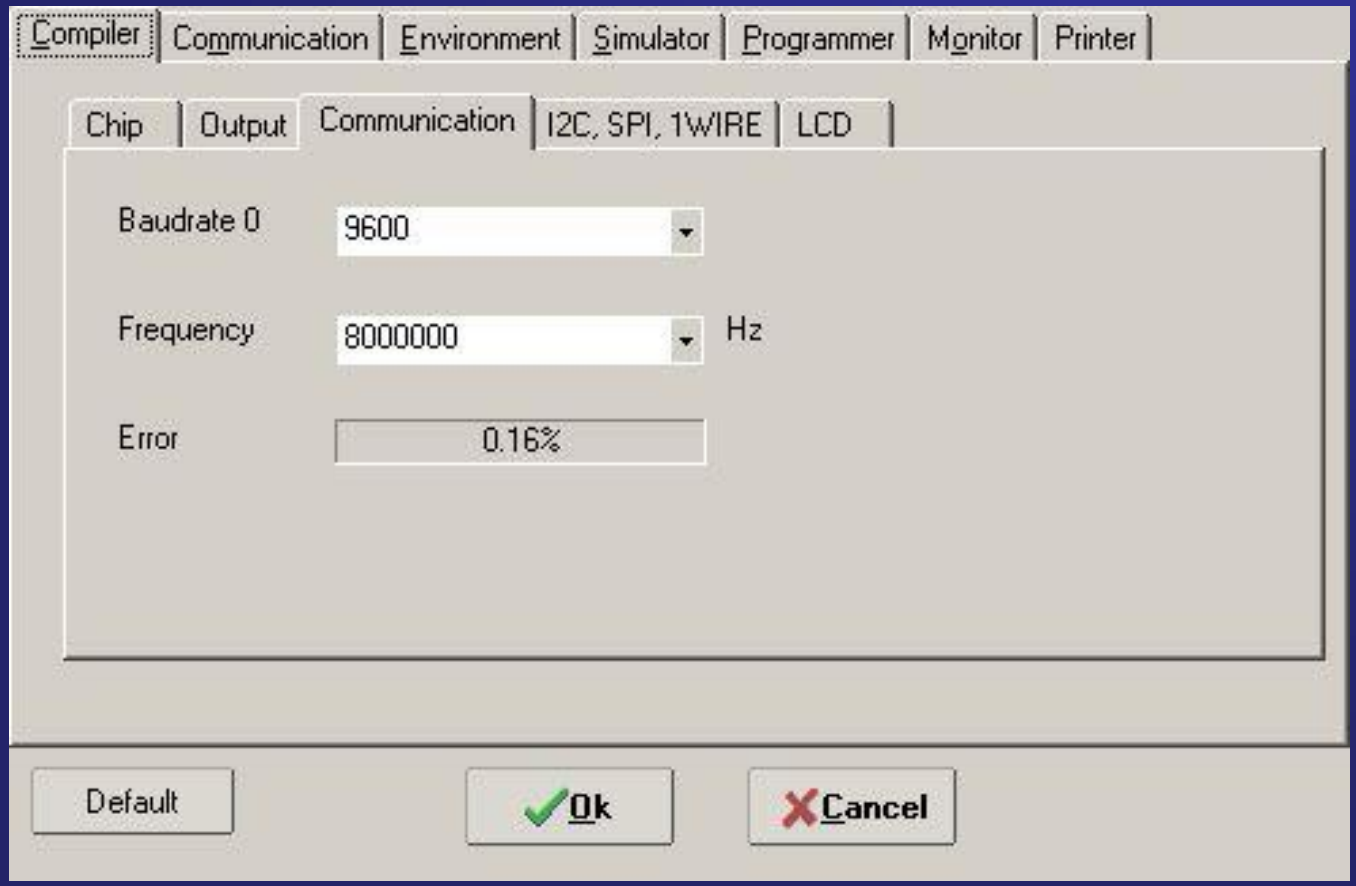
Oprócz linii danych w normie interfejsu przewidziano także kilka innych, które sterują przepływem danych. Do podstawowej komunikacji wystarczą jednak jedynie linie TxD oraz RxD, które podczas komunikowania się dwóch układów (np. mikrokontrolerów) powinny być skrzyżowane: wyjście TxD pierwszego układu powinno być dołączone do wejścia RxD drugiego układu, a wyjście TxD drugiego układu połączone z wejściem RxD pierwszego układu. Pozostałe sygnały sterujące RTS i CTS oraz DSR, DTR, DCD powinny zostać zapętlone.

Połączenie takie nazywa się null-modem i umożliwia transmisje asynchroniczna w dwóch kierunkach.

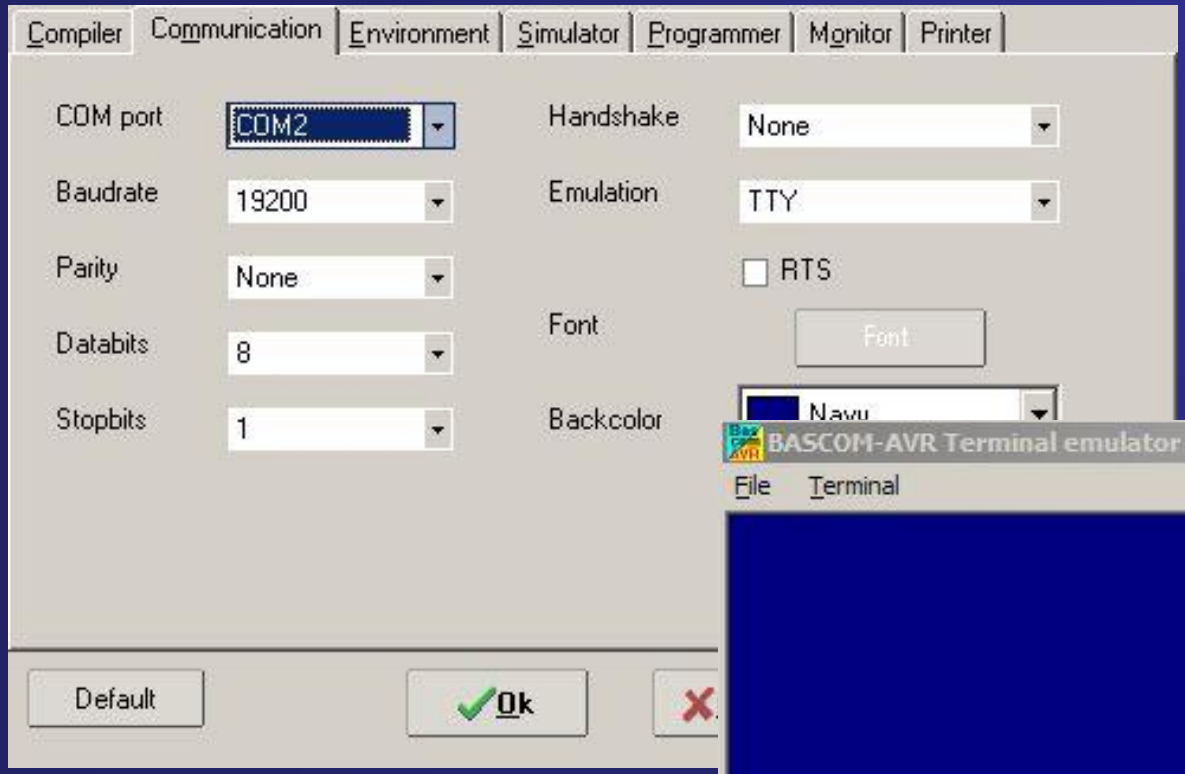


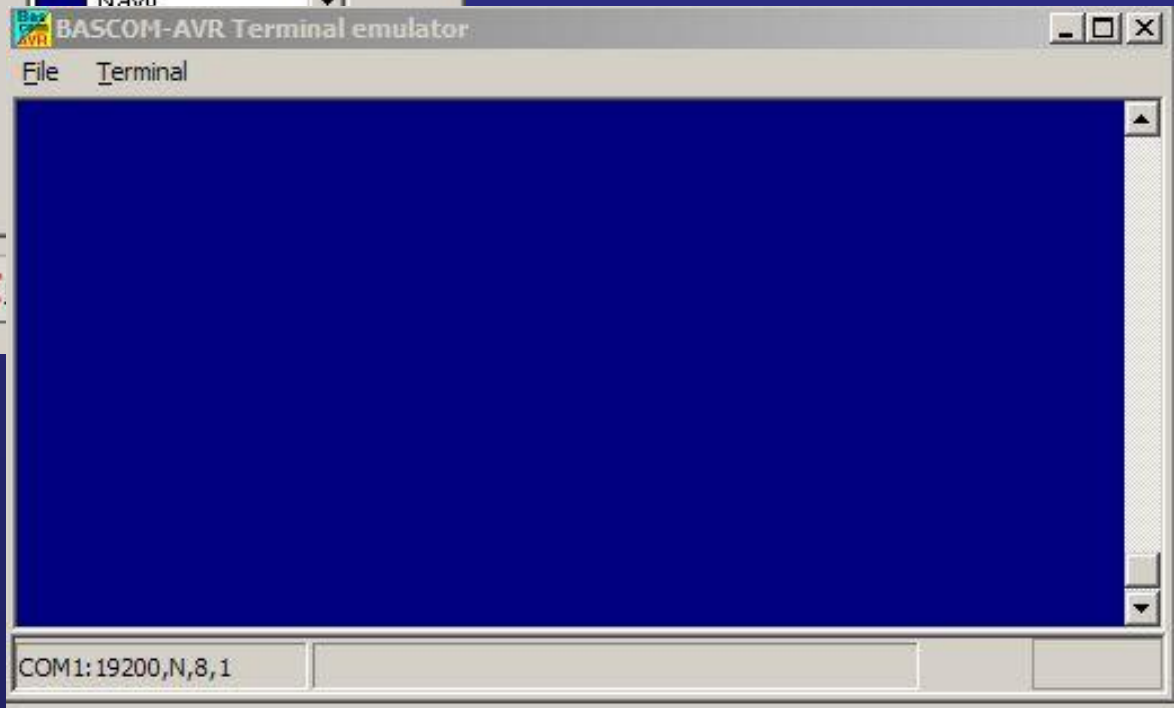


Konfiguracja interfejsu RS232 programu Bascom AVR



Konfiguracja terminala interfejsu RS232 programu Bascom AVR





Instrukcje Bascom AVR do komunikacji przez interfejs RS232

PRINT [zmienna | stała] [; [zmienna | stała]]

gdzie: zmienna, stała - dowolna stała lub zmienna

Przeznaczenie: Wysyła dane przez sprzętowy lub programowy UART lub zapisuje ciąg znaków do pliku

---

INPUT ["tekst_zachęty"] , zmienna1 [, zmiennan]

gdzie: tekst_zachęty opcjonalnie - tekst pojawiający się w oknie terminala, zmienna1, zmienna - zmienna której przypisana zostanie odebrana wartość,

Przeznaczenie: Pozwala na wprowadzanie danych, za pomocą zewnętrznego terminala.

ECHO ON | OFF

Przeznaczenie: Włącza i wyłącza echo dla instrukcji INPUT.

Do wprowadzania danych służy instrukcja INPUT, która wszystkie odebrane znaki przesyła z powrotem do terminala, by użytkownik widział co pisze. Gdy taka akcja nie jest wymagana lub w terminalu włączono lokalne echo, w programie można umieścić ECHO z parametrem OFF. Spowoduje to wyłączenie “odbijania” wpisywanych znaków.

Instrukcje Bascom AVR do komunikacji przez interfejs RS232

zmienna = INKEY( )

gdzie: zmienna - dowolna zmienna typu Byte, Integer, Word, Long lub Single

Przeznaczenie: Zwraca kod ASCII pierwszego znaku znajdującego się w buforze transmisji szeregowej.

---

zmienna = ISCHARWAITING()

gdzie: zmienna - dowolna zmienna typu Byte, Integer, Word, Long lub Single

Przeznaczenie: Zwraca wartość 1 gdy w buforze transmisji sprzętowego układu UART znajduje się oczekujący znak.

---

zmienna = WAITKEY()

gdzie: zmienna - zmienna tekstowa lub numeryczna, do której wpisany będzie kod ASCII odebranego znaku,

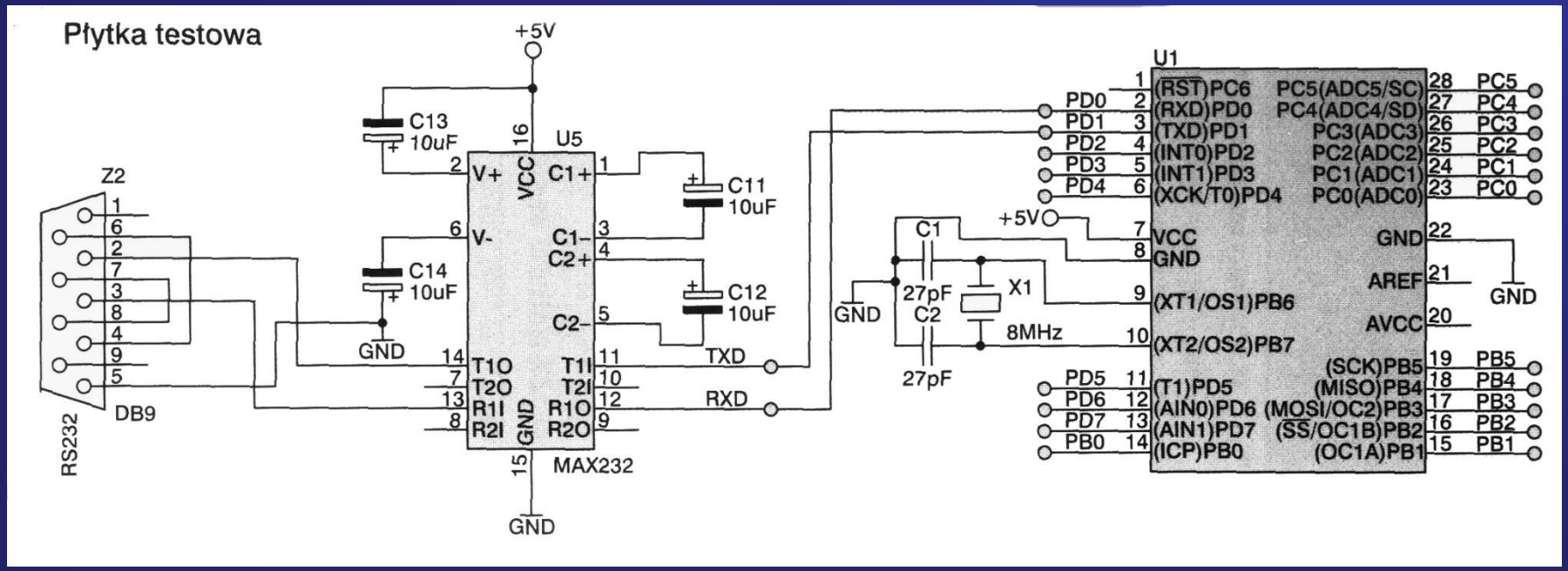
Przeznaczenie: Funkcja wstrzymuje działanie programu do czasu pojawienia się w buforze transmisji szeregowej odebranego znaku

Program 20

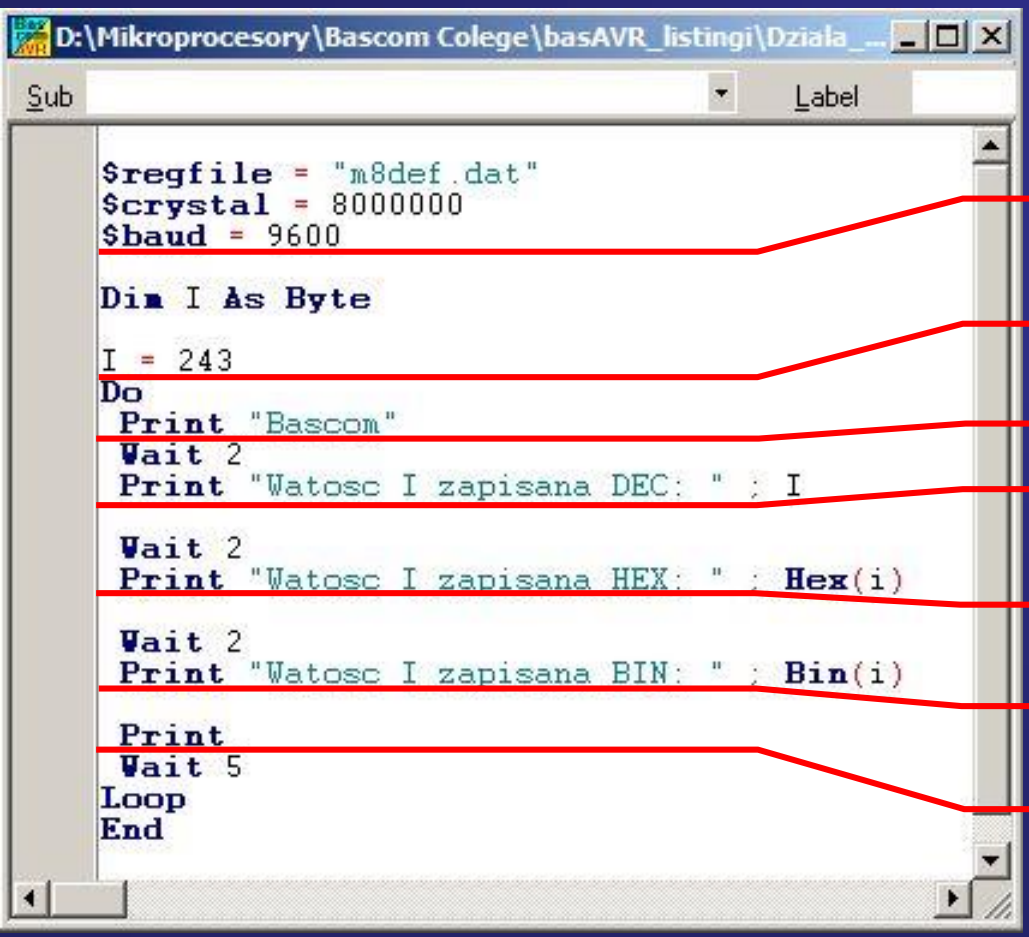
Nadawanie znaków przez sprzętowy interfejs RS232

Program 20

Schemat dołączenia do mikrokontrolera poprzez konwenter napięć (MAX232) do komputera



Program 20



informuje kompilator o prędkości transmisji

przypisanie zmiennej I wartości 243

wysłanie przez RS232 tekstu zawartego w cudzysłowie

wysłanie wartości zmiennej I

wysłanie wartości I w formacie szesnastkowym

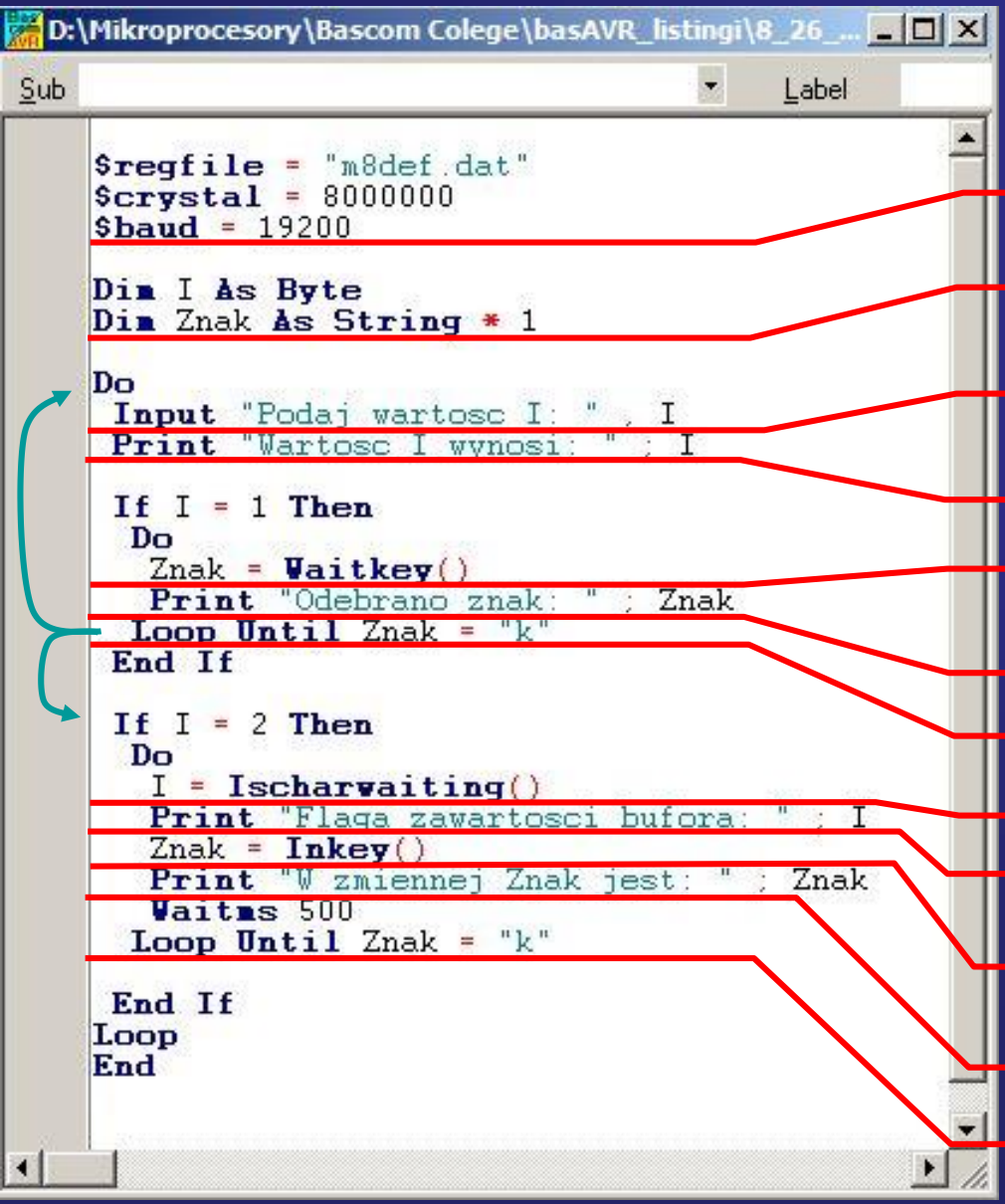
wysłanie wartości I w formacie binarnym

wstawienie linii odstępu pomiędzy wysyłanymi danymi

Program 21

Odbiór znaków przez sprzętowy interfejs RS232

Program 21



informuje kompilator o prędkości transmisji

definicja zmiennej Znak mogącej przechowywać 1 znak

wyświetlenie podanego tekstu oraz pobranie wartości przez RS232 do zmiennej I

wysłanie przez RS232 tekstu, po którym wysłana zostanie wartość zmiennej I

oczekiwanie na odebranie znaku przez RS232

wysłanie przez RS232 tekstu oraz odebranego znaku w Znak

jeśli odebrany znak to k, pętla jest opuszczana

sprawdzenie, czy w buforze jest znak do odczytania

wysłanie po tekście w cudzysłowu wartości zmiennej I

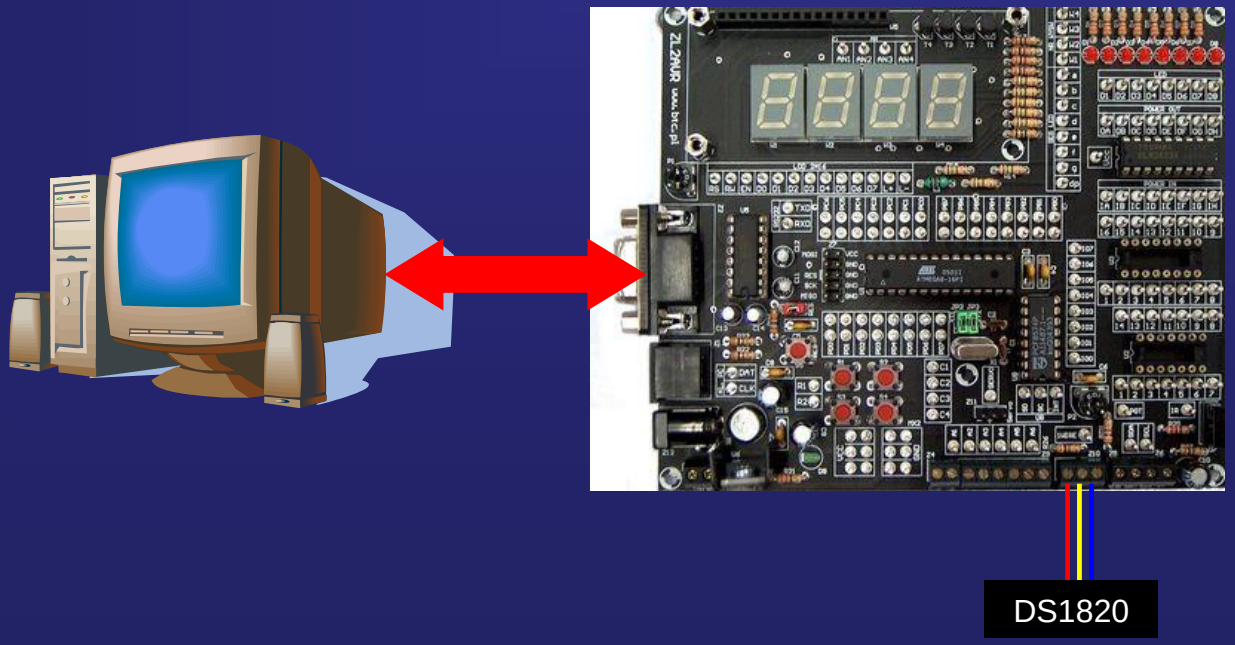
odebranie znaku z RS232, ale funkcja nie czeka na znak

wysłanie przez RS232 po tekście wartości zmiennej Znak

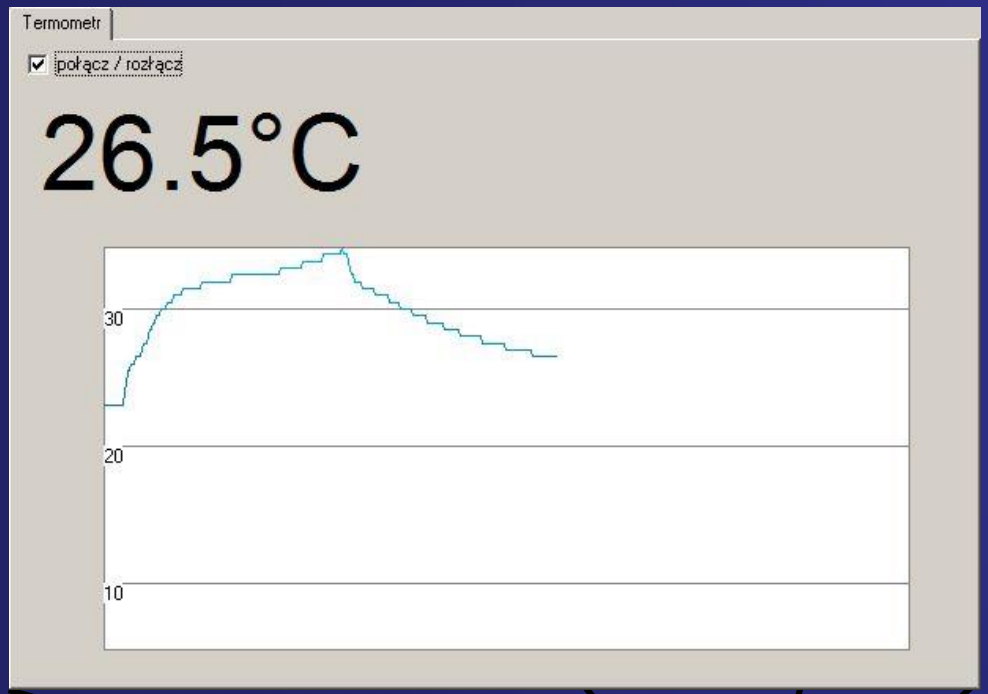
jeśli odebrany znak to k, pętla jest opuszczana

Zadanie specjalne !!!

Używając transmisji szeregowej RS232 wysłać informacje o mierzonej temperaturze czujnikiem DS1820 do komputera i odebraną informacje wyświetlić w okienku terminala

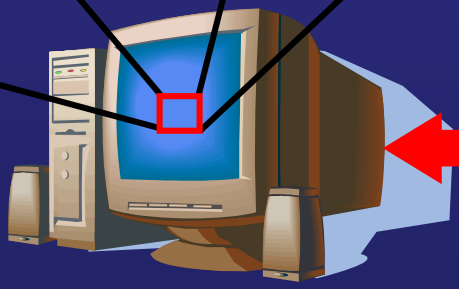


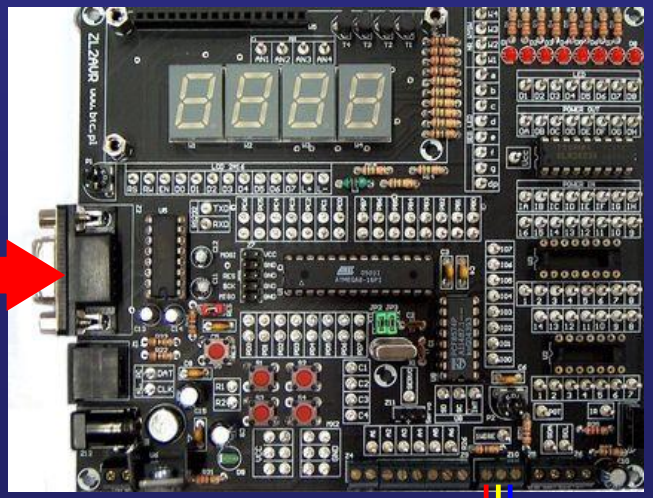
Zadanie specjalne !!!



Program archiwizacji temperatury

rs232_ds1820





DS1820

Program 22

D:\Mikroprocesory\Bascom Colege\basAVR_listingi\Dziala_zajecia\8_29_tem...

Sub | Label

```
$regfile = "m8def.dat"
$crystal = 8000000
$baud = 19200
Config Lcd = 16 * 2
Config Lcdpin = Pin , Db4 = Portc.3 , Db5 = Portc.2 , Db6
Config 1wire = Portb.0

Declare Sub Odcz_temp
Dim Temperatura(2) As Byte
Dim Cmd As Byte
Dim Znak As String * 1
Dim Tempst As Integer
Dim Tempdz As Integer

Deflcdchar 0 , 7 , 5 , 7 , 32 , 32 , 32 , 32 , 32

Do
 Call Odcz_temp

 Cls
 Lcd Znak ; Tempst ; "." ; Tempdz ; Chr(0) ; "C"

Cmd = Inkey()
  If Cmd = "G" Then
     Print Tempst ; "." ; Tempdz ;
     Print Chr(13) ; Chr(10);
  End If
Loop
End
```

Program 22

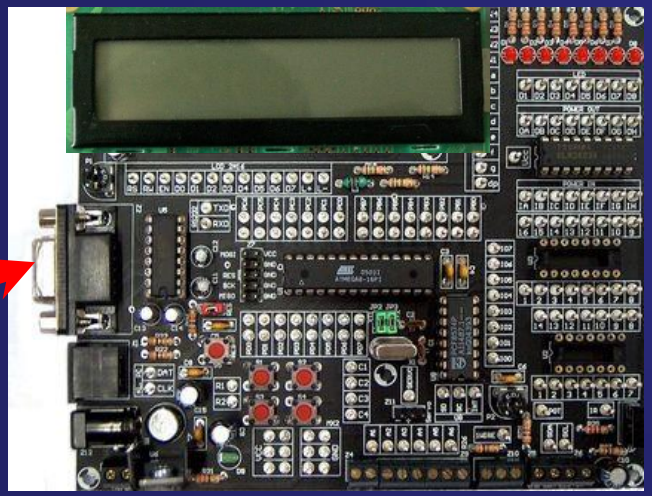
D:\Mikroprocesory\Bascom Colege\basAVR_listingi\Dziala_zajecia\8_29_tem...

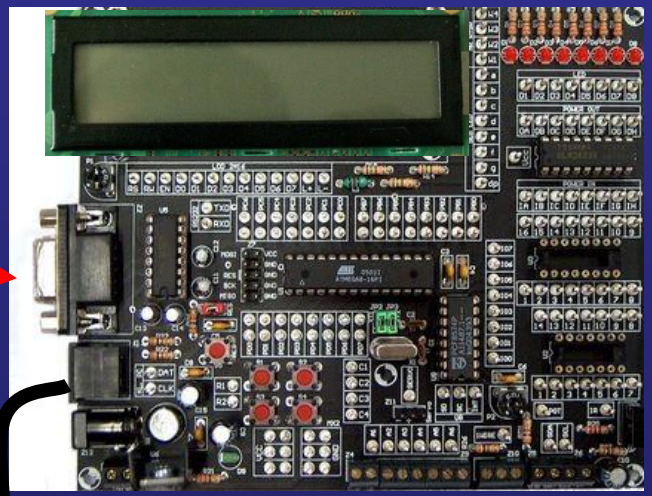
Sub | Label

```
Sub Odcz_temp

 1wreset
 1wwrite &HCC
 1wwrite &H44
 Waitms 750
 1wreset
 1wwrite &HCC
 1wwrite &HBE
 Temperatura(1) = 1wread(2)
 1wreset
 If Err = 1 Then
  Cls
  Lcd "Brak ukladu"
  Do
  Loop
 End If

If Temperatura(2) > 0 Then
          Temperatura(2) = 255 - Temperatura(2)
          Znak = "-"
       Else
          Znak = "+"
       End If
       Tempst = Temperatura(1) / 2
       Tempdz = Temperatura(1) Mod 2
       Tempdz = 5 * Tempdz

End Sub
```

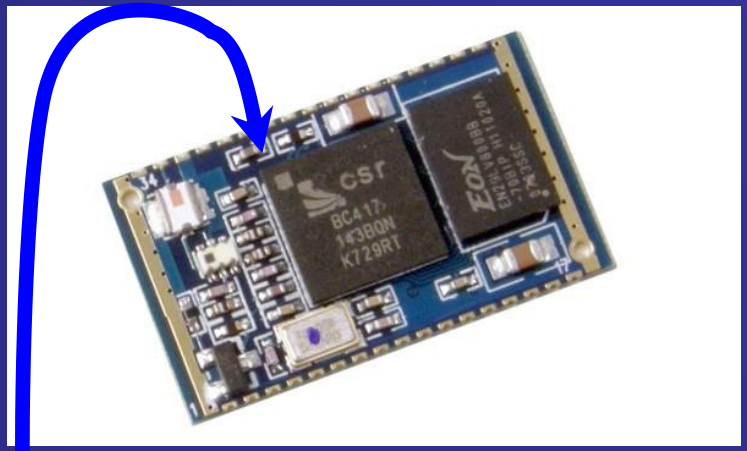
Zadanie specjalne !!!

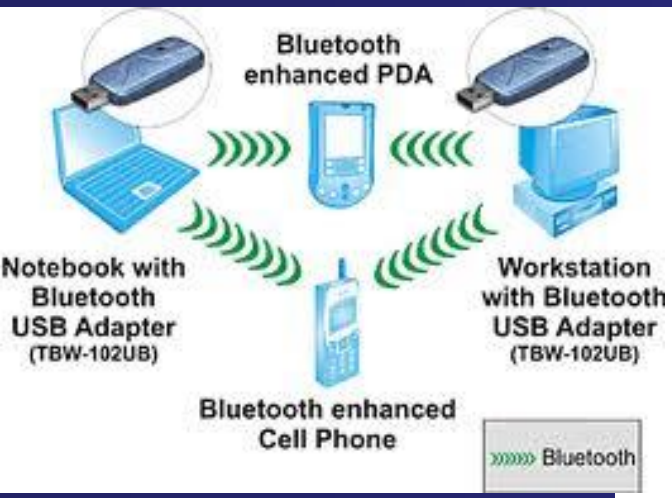
Używając klawiatury AT od PC i obsługi wyświetlacza LCD wysłać wiadomość do drugiego mikroprocesora przy pomocy transmisji szeregowej RS232

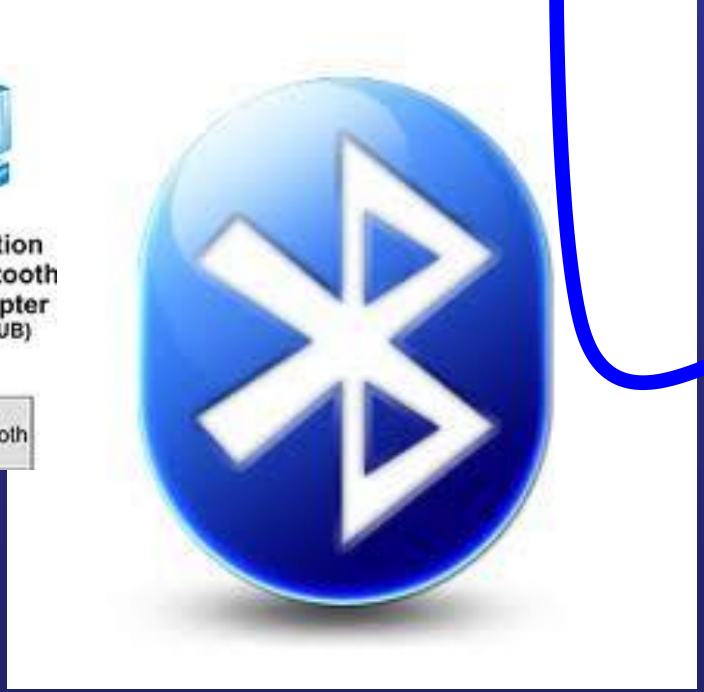


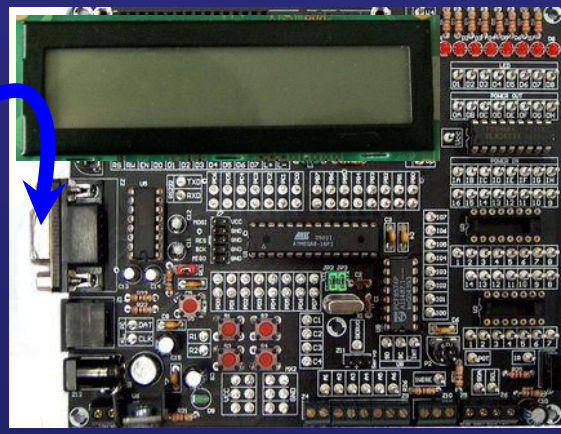


Transmisja bezprzewodowa Bluetooth !!!



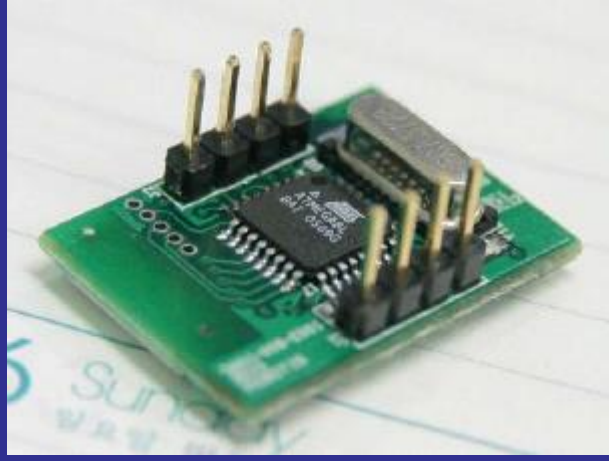


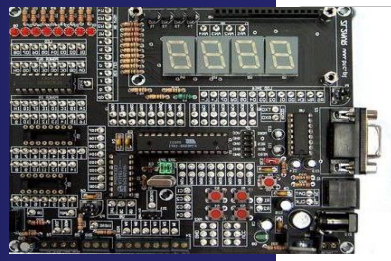


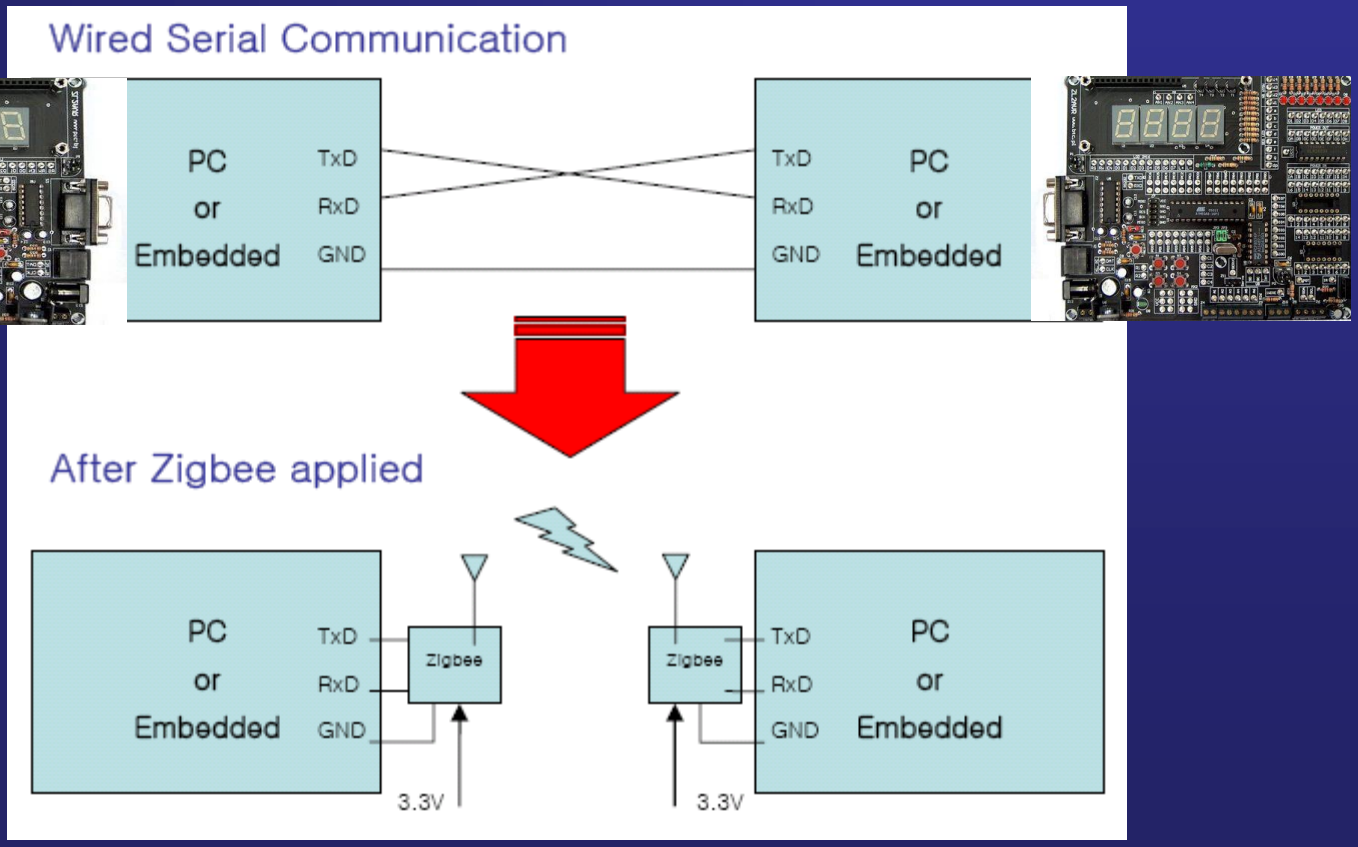


Bluetooth jest darmowym standardem bezprzewodowej komunikacji pomiędzy różnymi urządzeniami elektronicznymi. Zasięg komunikacji w otwartym terenie może wynieść nawet do 200m. Standard ten używa fal radiowych w paśmie ISM 2,4GHz.

Transmisja bezprzewodowa Zigbee !!!

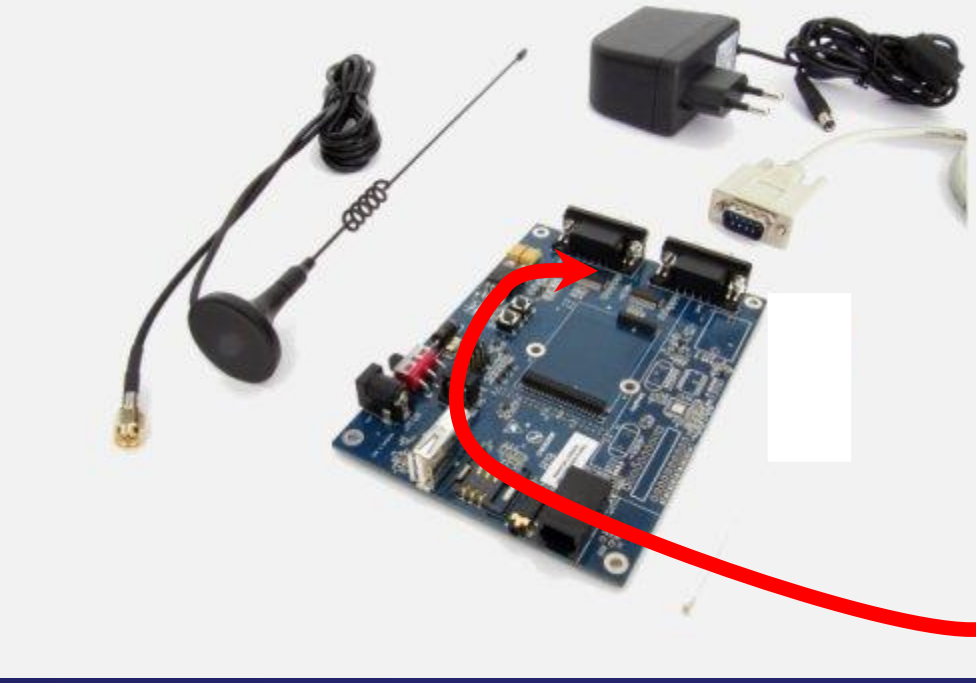


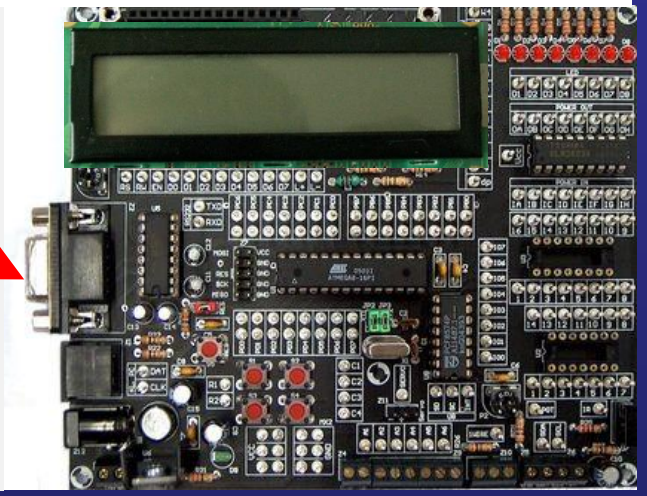




Używając transmisji szeregowej RS232 przy pomocy modułu Zigbee można w prosty sposób bezprzewodowo połączyć się innym systemem wbudowanym lub komputerem PC

Zestawy uruchomieniowe modemów GSM





Używając komend AT i transmisji szeregowej RS232 prosty system wbudowany może sterować modułem GSM