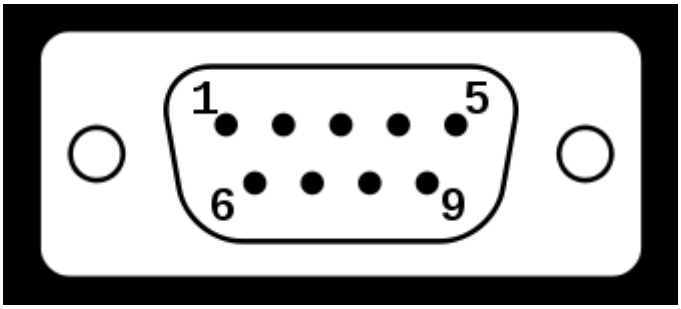
Laboratorium 3
Przesyłanie/Odbieranie danych poprzez UART.

Zadanie ma na celu zapoznanie się z najbardziej popularnym interfejsem do przesyłu danych jakim jest UART (universal asynchronous receiver/transmitter). Na tym laboratorium zostanie on wykorzystany wraz ze standardem RS-232 (wprowadzony w latach 1960'). Interfejs ten jest wykorzystywany w bardzo wielu zastosowaniach, obecnie bardzo często jak pomost pomiędzy USB a podłączanym urządzeniem (GPS, odbiorniki-piloty, kasy fiskalne, modemy, ...). Pomosty UART-USB najczęściej są pewną modyfikacją standardy i pracują najczęściej w zakresie zasilania od 0V do 3(5)V w stosunku do oryginału w zakresie od -15V do 15V. Jednak podstawowe założenia są takie same.

Interfejs w standardzie RS-232



Widok gniazda PC (męskiego) typu DE-9

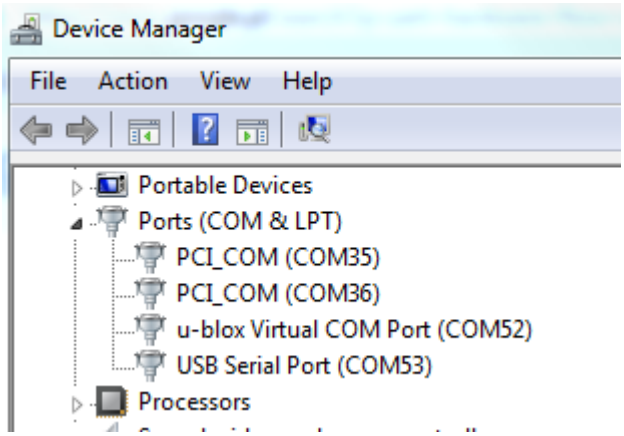
Numer 9 pin	Numer 25 pin	Kierunek	Oznaczenie	Nazwa angielska	Nazwa polska
1	8	DCE – > DTE	DCD	Data Carrier Detected	sygnał wykrycia nośnej
2	3	DCE – > DTE	RxD	Receive Data	odbiór danych
3	2	DCE < – DTE	TxD	Transmit Data	transmisja danych
4	20	DCE < – DTE	DTR	Data Terminal Ready	gotowość terminala
5	7	DCE – DTE	GND	Signal Ground	masa
6	6	DCE – > DTE	DSR	Data Set Ready	gotowość "modemu"
7	4	DCE < – DTE	RTS	Request to Send Data	żądanie wysyłania
8	5	DCE – > DTE	CTS	Clear to Send Data	gotowość wysyłania
9	22	DCE – > DTE	RI	Ring Indicator	wskaźnik dzwonka
	9-19; 21; 23-25		NC		niewykorzystane

https://pl.wikipedia.org/wiki/RS-232

W większości przykładowych zastosowań wykorzystywane są dzisiaj tylko złącza RxD (nazywane też RX) oraz złącze TxD (nazywane też TX) oraz masa (GND). Ten minimalny zestaw pozwala na przesyłanie danych w trybie programowego sterowania przepływem (software flow control), możliwe jest też sprzętowe sterowanie przepływem (Hardware Flow Control) z wykorzystaniem złącz RTS i CTS. To drugie jest wskazane w przypadku kiedy jedno urządzenie może nie nadążać za drugim.

Na laboratorium należy stworzyć dwa programy (w zasadzie jeden uruchomiony dwa razy).

Powinny one umożliwiać ustalenie danych przesyłu (na pewno szybkość, i rodzaj kontroli przepływu). Jedna instancja programu powinna połączyć się z jednym interfejsem USB-UART (widzianym w systemie jako COM) a druga instancja z drugim interfejsem widzianym jako COM. W menadżerze urządzeni można zobaczyć, które numery zostały przydzielone przez system.



Do obsługi portu szeregowego można wykorzystać klasę Serial z platformy .NET (najszybsze rozwiązanie w przypadku komputerów na laboratorium). Do programu należy wprowadzić opcjonalne opóźnienie aby móc przetestować typy sterowania przepływem. Programy powinny między sobą przesyłać dane (w uproszczeniu mogą to być kolejne liczby). W opisanym przypadku nawet wskazane jest aby programy były uruchomione na dwóch różnych komputerach.

Zadania:

1) Połączenia należy zrealizować dla programowego kontrolowania przepływu
RxD <----> TxD
TxD <----> RxD
GND <----> GND

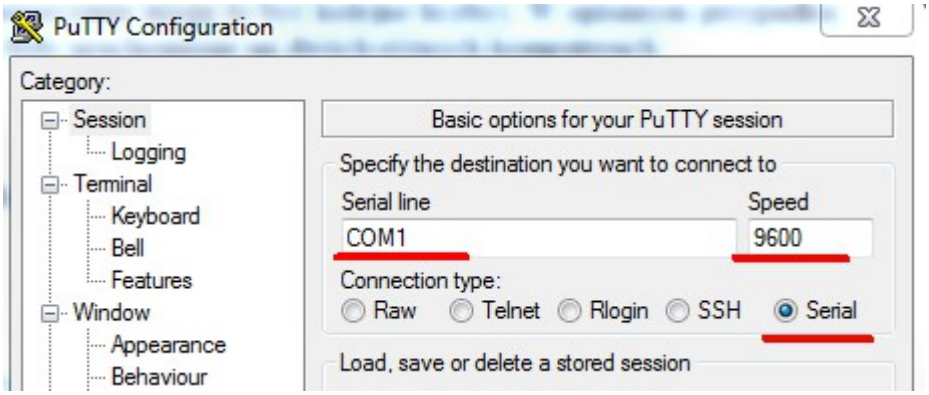
2) oraz sprzętowego kontrolowanie przepływu
RxD <----> TxD
TxD <----> RxD
CTS <----> RTS
RTS <----> CTS
GND <----> GND

3) Z otrzymanych rezultatów należy sporządzić krótkie zestawienie/opis

Interfejsy oparte są o układy FTDI232 i takie sterowniki też powinno się zainstalować (ze względu na pojawienie się podrób układów FTDI te zestawy na pewno pracują z wersją 2.12.0.0).

UWAGA: do odbioru danych (przypadek SerialPort z .NET) należy użyć zdarzenia DataReceived. Do przesyłania danych wykorzystujemy funkcję write. Należy przetestować zarówno pisanie bajt po bajcie jak i zapis całego bloku (1k, 100k, 1M). W procedurze DataReceived można zasymulować opóźnienie (np. poprzez wywołanie Thread.sleep)

Do przetestowania połączeni można użyć terminalu putty, na dwóch komputerach połączonych poprzez interfejs RS-232 wywołujemy połączenie z tą samą konfiguracją:



Po uzyskaniu połączenia pisząc w oknie jednej konsoli, tekst powinien pojawiać się w drugiej.