



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Program Rozwojowy
Politechniki Warszawskiej

EFES
Europejski Fundusz Społeczny



Zadania realizowane przez Wydział Elektryczny PW

Wprowadzenie do środowiska Visual Studio cz. 2 (ćwiczenie wykonywane zdalnie)

Wykorzystanie Wirtualnego portu COM

1.1 **Cel ćwiczenia:** Celem ćwiczenia jest zaznajomienie studenta z tematyką tworzenia aplikacji z wykorzystaniem oprogramowania Visual Studio. Największy nacisk zostanie położony na wykorzystanie wirtualnych portów COM oraz symulatorów urządzeń. Symulatory te mogą wspomóc proces tworzenia aplikacji poprzez symulowanie softwarowe urządzenia rzeczywistego. Takie podejście pozwoli na pisanie programów bez potrzeby dostępu do fizycznego sprzętu (czasem kosztownego).

1.2 **Plan ćwiczenia** przedstawia się następująco:

- a) Zapoznanie się z pojęciem wirtualnego portu COM.
- b) Konfiguracja wirtualnego portu COM z wykorzystaniem programu com0com.
- c) Przykład komunikacji pomiędzy portami z wykorzystaniem terminala.
- d) Omówienie przykładowego symulatora GPS.
- e) Wykorzystanie symulatora GPS i terminala.
- f) Zadania do samodzielnego rozwiązania.

1.3 **Wykonanie ćwiczenia**

a) **Zapoznanie się z pojęciem wirtualnego portu COM.**

Port szeregowy (ang. Serial Port) to port komputerowy, przez który dane są przekazywane w formie jednego ciągu bitów. Starsze komputery wykorzystywały porty szeregowo potocznie zwane portami COM do komunikacji komputera z urządzeniami zewnętrznymi. Komputery przemysłowe do tej pory posiadają kilka a nawet kilkanaście portów COM (najczęściej interfejsy RS232) jednak w nowoczesnych komputerach domowych już rzadko znajdziemy takie rozwiązanie. Rozwiązanie to zostało wyparte przez nowoczesne interfejsy na przykład USB. Wirtualny port COM to stworzony programowo driver symulujący taki interfejs. Nawet interfejsy USB wykorzystują



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



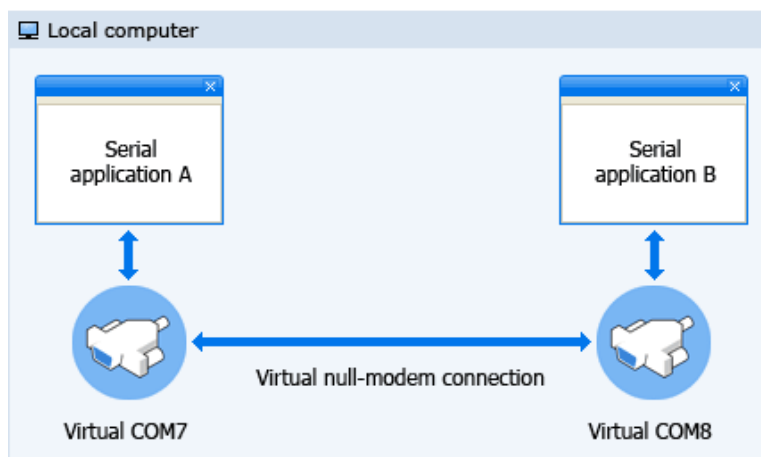
PROGRAM ROZWOJOWY
POLITECHNIKI WARSZAWSKIEJ

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



wirtualne porty COM i to za ich pomocą możemy komunikować się z zewnętrznym urządzeniem.

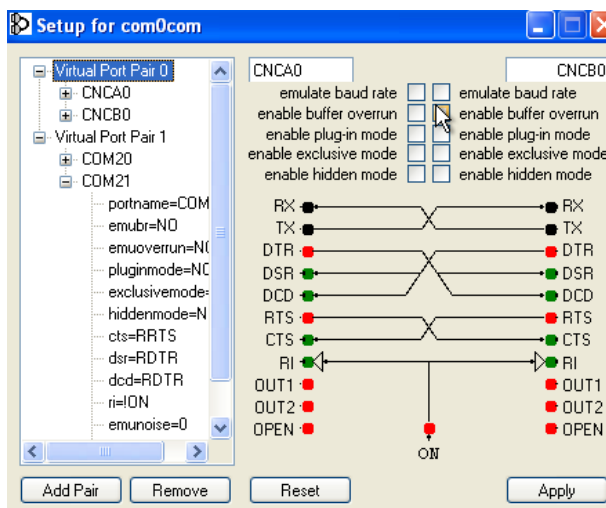
Możemy jednak wykorzystać pomysł wirtualnego portu COM i sparować dwa porty ze sobą. Oznacza to, że to co wyślemy na Port COM 7 będzie widoczne na porcie COM 8 (patrz rysunek niżej). Lub odwrotnie to co na COM 8 będzie widoczne na porcie COM 7.



W ten sposób możemy napisać aplikacje, która wysyła lub odbiera dane z portu COM7 a za pomocą tego połączenia dane przesyłane są do portu COM8 (które może być naszym programowym symulatorem urządzenia np. GPS).

b) Konfiguracja wirtualnego portu COM z wykorzystaniem programu com0com.

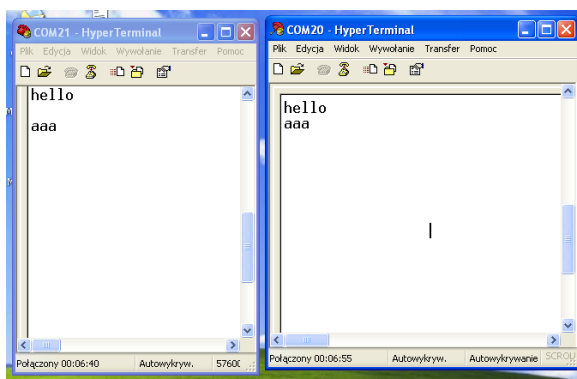
Istnieje bardzo wiele aplikacji umożliwiających tworzenie wirtualnego portu COM. My jednak będziemy używali darmowego programu com0com. Programu tego będziemy chcieli użyć do sparowania dwóch portów COM ze sobą. Program można pobrać ze strony: <http://sourceforge.net/projects/com0com/>. Jeśli nie masz zainstalowanego programu powinieneś wykonać poniższe czynności które znajdują się w instrukcji instalacji: <http://support.flexradio.com/Downloads.aspx?id=218> (lub plik w zasobach KSW). Wypakuj pliki do katalogu a następnie uruchom plik instalacyjny. Po przejściu instalacji trzeba przejść do skonfigurowania (w naszym przypadku sparowania) portów COM. Sprawdź jak to się robi w pomocy programu. Utwórz dwa sparowane ze sobą porty COM za pomocą linii poleceń. W naszym przypadku będą to porty COM20 i COM21. W celu sprawdzenia konfiguracji sprawdź ustawienia w menu **Start->Programy->com0com->Setup**.



Powinieneś zobaczyć panel konfiguracyjny i sparowane ze sobą porty COM u nas COM20 i COM21.

c) Przykład komunikacji pomiędzy portami z wykorzystaniem terminala.

Teraz przyszła pora na sprawdzenie naszych sparowanych portów COM. W tym celu posłużymy się terminalami z systemu Windows. Wybieramy **Start->Programy->Akcesoria->Hyper Terminal** a następnie konfigurujemy ustawienia terminala na port COM20. Postępujemy ponownie tak samo tyle, że tym razem nawiązujemy połączenie z portem COM21. Ponieważ nasze porty COM są ze sobą sparowane wszystko co napiszemy w okno z portem COM21 powinno być wyświetlane w oknie z otworzonym portem COM20 i na odwrót.



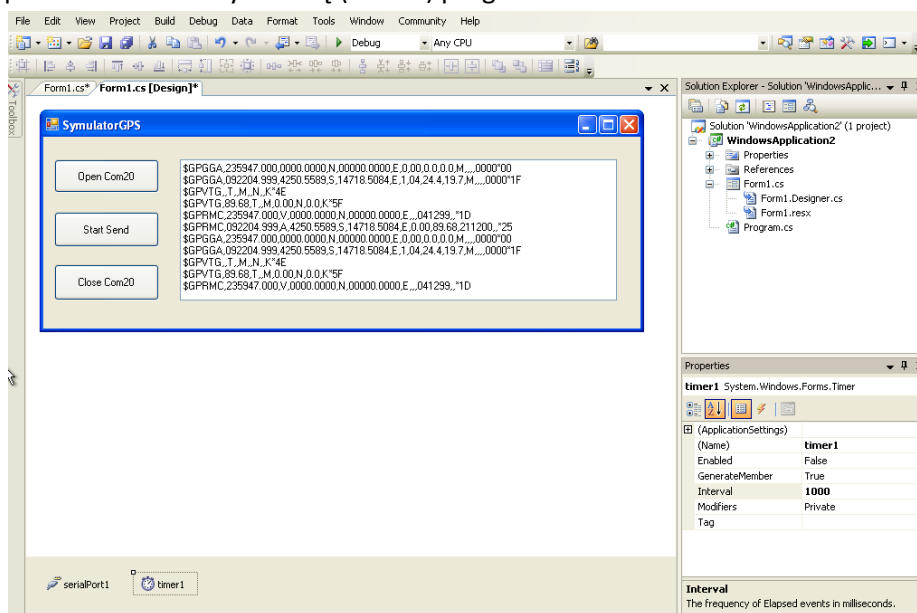
d) Omówienie przykładowego symulatora GPS.

Można sobie teraz wyobrazić, że będziemy dysponowali programem, który będzie zachowywał się jak prawdziwy odbiornik GPS a swoje dane będzie wysyłał na wirtualny port COM20. My natomiast będziemy odbierać te dane na sparowanym porcie COM21. Zaletami takiego rozwiązania są między innymi:

- nie potrzebujemy sprzętu, czasem bardzo drogiego
- możemy testować nasze aplikacje w domu a wykorzystać sprzęt, gdy nasz program będzie już dopracowany w szczegółach
- jest to bardzo wygodne, często uruchomienie sprzętu wymaga wielu czasochłonnnych zabiegów

Zajmiemy się teraz omówieniem prostego symulatora GPS. Symulator taki w odpowiednich odstępach czasu (tak jak jego sprzętowy odpowiednik) podaje informacje na temat aktualnej pozycji GPS (Więcej na ten temat powinieneś dowiedzieć się na wykładzie). Informacje te wysyłane są za pomocą linii tekstu w którym znajdują się między innymi dane na temat pozycji, wysokości i wiele innych (patrz opis GPS).

Program ten jest już stworzony i możesz znaleźć go w zasobach KSW (Symulator GPS). Działanie aplikacji polega na wysłaniu tekstów (stringów), które znajdują się w polu textBox1 na port COM20 (ustawienia SerialPort1). Wysyłanie tych tekstów odbywa się co pewien ustalony czas w komponencie timer1.interval. Po załadowaniu programu powinieneś zobaczyć formę (Form1) programu.



Na formie programu znajdują się trzy przyciski które służą odpowiednio do:

- Otwarcia portu COM20
- Wysłania Tekstu znajdującego się w textBox1
- Zamknięcia portu COM20.

Teraz powinieneś przyjrzeć się dokładnie zdarzeniom (Events) jakie wykonywane są podczas naciśnięcia przycisków (Button) lub podczas wywołania timera. Poświęć na to troszkę czasu aby całkowicie zrozumieć działanie programu. Jest tu wiele przydanych elementów, które mogą się przydać podczas pisania kolejnych aplikacji.

Otworzenie portu Com:

```
private void button2_Click(object sender, EventArgs e)
{
    serialPort1.Open();
}
```

Pobranie losowo tekstu wpisanego w textBox1 i wysłanie go na port COM:

```
private void timer1_Tick(object sender, EventArgs e)
{
    textBox1.WordWrap = false;
    int requiredLine, totalLines;
    totalLines = 10;

    losuj losowacz = new losuj();
    requiredLine = losowacz.zakres(1,10);

    string[] lines = textBox1.Text.Split("\r".ToCharArray(), totalLines);

    serialPort1.Write(lines[requiredLine - 1]+"\r");
}
```

Zamknięcie portu COM:

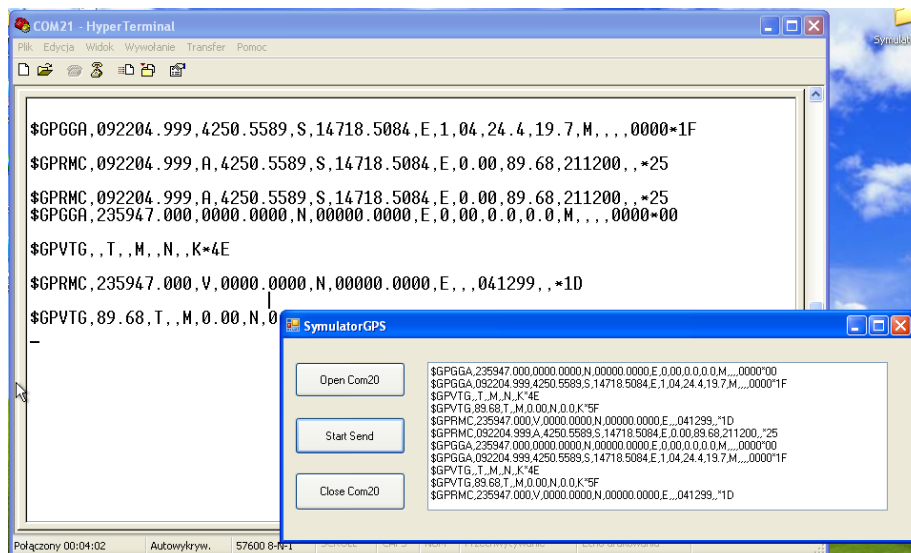
```
private void button3_Click(object sender, EventArgs e)
{
    timer1.Enabled = true;
}

private void button4_Click(object sender, EventArgs e)
{
    timer1.Enabled = false;
    serialPort1.Close();
}
```

e) Wykorzystanie symulatora GPS i terminala.

Jeśli już rozumiesz ideę wirtualnego portu COM oraz symulatora GPS. Przyszła pora na sprawdzenie aplikacji emulującej GPS w praktyce. Dysponując programem com0com stworzyliśmy wirtualne porty COM20 i COM21, które są ze sobą sparowane. Teraz podłączamy nasz symulator do portu COM20 natomiast otwieramy dodatkowo terminal ustawiony na port COM21.

Włączamy nasz symulator GPS. Wybieramy przycisk OpenCom20 a następnie Start Send i w oknie terminala powinniśmy zobaczyć pojawiające się teksty takie jak otrzymamy z prawdziwego urządzenia GPS.



f) Zadania do samodzielnego rozwiązania.

Twoim kolejnym zadaniem będzie napisanie programu który odbiera dane z urządzenia GPS. Oczywiście jak się domyślasz możesz obejść się bez urządzenia i wykorzystać wcześniej omówiony symulator. W tym celu wykorzystaj komponenty takie jak SerialPort, Timer.

Literatura/WWW:

[1] <http://sourceforge.net/projects/com0com/>

[2] <http://support.flexradio.com/Downloads.aspx?id=218>