Работа с COM-портами в .NET (C#)

Для работы с последовательными портами в .NET еще с ранних версий существует весь необходимый функционал. По большей части он заключен в использовании класса SerialPort, который располагается в пространстве имен System.IO.Ports. Чтобы было удобнее пользоваться классом SerialPort, можно сразу подключить необходимое пространство имен с помощью строчки:

using System.IO.Ports;

Получение списка COM-портов

Для получения списка всех COM-портов компьютера можно воспользоваться статическим методом GetPortNames класса SerialPort:

string[] portnames = SerialPort.GetPortNames();

В результате этого вызова в массиве portnames будет содержаться список всех COM-портов локальной машины.

Открытие COM-порта. Настройка

Перед использование конкретного COM-порта нужно сначала создать объект класса SerialPort с указанием имени порта и параметров скорости, четности и т.д. Вот пример открытия порта COM1 на скорости 9600 без контроля четности с 8-мью битами данных и 1-им стоп-битом.

SerialPort port = new SerialPort( ″COM1″ , 9600, Parity.None, 8, StopBits.One);

После создания объекта SerialPort нужно открыть порт:

port.Open();

В результате вызова этого метода может возникнуть исключение типа IOException. Например, со следующим описанием: *Порт 'COM1' не существует*. Для отправки данных в COM-порт служит метод Write. Пример кода, отправляющего в открытый порт 5 байт:

byte[] data = { 0, 1, 2, 1, 0 };

port.Write(data, 0, data.Length);

,где первый параметр - массив с данными, второй - позиция начала данных, третий - длина данных.

С чтением данных из COM-порта тоже все просто. Для этого служит метод Read. Пример использования:

byte[] data = new byte[256];

port.Read(data, 0, data.Length);

Этот набор инструкций считает из COM-порта блок данных максимального размера в 256 байт и запишет результат в массив data. Для побайтового чтения существует метод ReadByte

int databyte = port.ReadByte();

После того, как COM-порт нам уже не нужен его необходимо закрыть используя метод Close:

port.Close();

В случае если COM-порт уже используется другим процессом, то при попытке его открыть мы получим исключение:**Невозможно открыть com порт**. Чтобы определить, кто именно держит порт, нужно воспользоваться утилитой для мониторинга ресурсов процессов.