
Subject: SerialPort & SerialPortListener class
Posted by [omari](#) on Sat, 12 Mar 2016 13:05:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

SerialPort with non blocking read/write, and can show setting dialog when flagGUI is defined

SerialPortListener run on separate thread when flagMT is defined

Example of use:

SerialPort:

```
#include <Core/Core.h>
#include <SerialPort/SerialPort.h>

using namespace Upp;

CONSOLE_APP_MAIN
{
    SerialPort rs;

    rs.Settings().Baud9600().ParityOdd().StopBits1().DataBitsEight();

    rs.Open ( "Com1" );

    rs.Write("hello");

    if(rs.WaitReply())
    {
        Cout() << rs.Available();
        Cout() << rs.ReadAll();
    }

    rs.Write("quit");

    rs.Close();

    ReadStdIn();
}
```

SerialPortListener:

```
#include <SerialPort/SerialPort.h>

using namespace Upp;
```

```

bool quit = false;

String ProcessCommande(String& request)
{
    String response = "";
    request = ToLower(request);

    if(request == "help")
    {
        response = " time date quit help";
    }
    else
    if(request == "quit")
    {
        quit = true;
    }
    else
    if(request == "time")
    {
        response = AsString(GetSysTime());
    }
    else
    if(request == "date")
    {
        response = AsString(GetSysDate());
    }
    else
    {
        response = request + " : unknown request";
    }

    response << "\r\n";

    return response;
}

bool OnRequest(String request, String& response)
{
    static String buf = "";
    static int count = 0;

    // simple protocol : each frame has 4 bytes
    for(int i = 0; i < request.GetCount(); i++)
    {
        buf.Cat(request[i]);
        count++;
        if(count == 4)

```

```
{
  response << ProcessCommande(buf);
  count = 0;
  buf = "";
}
}

return quit;
}

CONSOLE_APP_MAIN
{
  SerialPortListener listener;
  listener.PortSettings().Baud9600();
  listener.WhenDataReceived = callback(OnRequest);
  listener.Start("com2");

  listener.Wait();
}
```

File Attachments

1) [SerialPort.7z](#), downloaded 286 times
