
Subject: Problem using boost library

Posted by [Giorgio](#) on Thu, 19 Jan 2017 10:30:49 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi there,

I have to write a small console application that every seconds polls a device attached to a serial port and saves what is returned in a database.

Since is my first attempt using serial device, first thing I did was to understand how to communicate with the serial port. After a long quest I found the boost library and a tutorial that helped me a lot. I created a test application that sends a message to the serial device and print on screen the answer. This test application was written in UPP using an empty package, so I guess is pretty much standard C++.

```
//main.cpp
#include <iostream>
#include <boost/thread.hpp>

//#include "AsyncSerial.h"
#include "BufferedAsyncSerial.h"

using namespace std;
using namespace boost;

int main(int argc, char* argv[])
{
    try {

        BufferedAsyncSerial tico("COM5",
            9600,
            boost::asio::serial_port_base::parity(boost::asio::serial_port_base::parity::even), //Parity even
            boost::asio::serial_port_base::character_size(8), //8bit (?)
            boost::asio::serial_port_base::flow_control(boost::asio::serial_port_base::flow_control::none),
            //Flow control none
            boost::asio::serial_port_base::stop_bits(boost::asio::serial_port_base::stop_bits::one)); //Stop
        bits 1

        for(int i = 0; i<5; i++) {
            tico.writeString("TAV R\r");

            //Simulate doing something else while the serial device replies.
            //When the serial device replies, the second thread stores the received
            //data in a buffer.
            this_thread::sleep(posix_time::seconds(1));
            cout<<tico.readString()<<endl;
        }
    }
}
```

```

        tico.close();

    } catch(boost::system::system_error& e)
    {
        cout<<"Error: "<<e.what()<<endl;
        return 1;
    }
}

```

That is the main.cpp file, if needed I can provide the BufferedAsyncSerial and AsyncSerial libraries. It compiles in UPP and successfully runs. As I said, this is just to test the serial device communication.

After that, since I have to store data in a database, I created a Core Console Project in UPP. In that project I added the above mentioned code. The project will mainly consist in a class that writes data retrieved from the serial device into the db. But in the first test below, I do not actually store data in the db: I am just trying to reproduce the application that I wrote in a Core Console Project.

```
#include "ProvaSP.h"
```

```
/*****For the serial port*****/
```

```
#include <iostream>
```

```
#include <boost/thread.hpp>
```

```
//#include "AsyncSerial.h"
```

```
#include "BufferedAsyncSerial.h"
```

```
using namespace std;
```

```
using namespace boost;
```

```
/******
```

```
const int NUM_SECONDS = 1;
```

```
CONSOLE_APP_MAIN
```

```
{
```

```
    try {
```

```
        BufferedAsyncSerial tico("COM5",
```

```
        9600,
```

```
        boost::asio::serial_port_base::parity(boost::asio::serial_port_base::parity::even), //Parity even
```

```
        boost::asio::serial_port_base::character_size(8), //8bit (?)
```

```
        boost::asio::serial_port_base::flow_control(boost::asio::serial_port_base::flow_control::none),
```

```
        //Flow control none
```

```
        boost::asio::serial_port_base::stop_bits(boost::asio::serial_port_base::stop_bits::one));
```

```
        //Stop bits 1
```

```

while(true)
{
tico.writeString("TAV R\r");
    this_thread::sleep(posix_time::seconds(NUM_SECONDS));
cout<<tico.readString()<<endl;
}
    tico.close();
} catch(boost::system::system_error& e)
{
    cout<<"Error: "<<e.what()<<endl;
    //return 1;
}
}

```

As you can notice, the code is barely the same, but when I compile I get the following errors:

ProvaSP.cpp

```

C:\local\boost_1_63_0\boost\asio\io_service.hpp(604): error C2208:
'boost::asio::detail::winsock_init<2,0>': tipo non utilizzato da alcun membro definito
C:\local\boost_1_63_0\boost\asio\detail\impl\socket_ops.ipp(1990): error C2039:
'WideCharToMultiByte': non è un membro di "global namespace"
C:\local\boost_1_63_0\boost\asio\detail\impl\socket_ops.ipp(1990): error C2065: 'CP_ACP':
identificatore non dichiarato
C:\local\boost_1_63_0\boost\asio\detail\impl\socket_ops.ipp(1990): error C3861:
'WideCharToMultiByte': identificatore non trovato
C:\local\boost_1_63_0\boost\asio\detail\impl\socket_ops.ipp(2197): error C2039:
'MultiByteToWideChar': non è un membro di "global namespace"
C:\local\boost_1_63_0\boost\asio\detail\impl\socket_ops.ipp(2197): error C2065: 'CP_ACP':
identificatore non dichiarato
C:\local\boost_1_63_0\boost\asio\detail\impl\socket_ops.ipp(2197): error C3861:
'MultiByteToWideChar': identificatore non trovato

```

For some reason UPP compiles successfully a standard project containing boost library, but has no success if the same code is in a Console Core Project.

Attached, the Console Core Project, of course you need to have boost libraries in your system to try to compile it.

Prior to use boost library, I checked out this thread, but I did not have success in communicating with the serial device, maybe because in the libraries attached to the thread there is no way to set parity, control flow etc, but only baud rate and serial port number.

Please, this is a very important and urgent matter to me, I will appreciate any kind of advice, including RTFD or an alternative to boost library.

Regards,
Gio

File Attachments

1) [ProvaSP2.zip](#), downloaded 247 times
