

---

# RtxSerial.dll with WM 6.1

L0cke

10.11.2011

## Table of Contents

1. Problem description .....	1
2. rtxSerial for WinCE .....	1
2.1. Getting WinCE rtxserial sources .....	1
2.2. Getting embedded visual c++ and additional software .....	2
2.3. Setting up rtxserial WinCE for build .....	2
2.4. Building dll .....	5

*For my wife Agnieszka*

## 1. Problem description

At my work there was a problem, that we had a mobile device with Windows Mobile 6.1, and on this device there was a RFID module, which was connected to the serial port. So, normally using Java, you had to use rtxSerial ports. The problem was, that there was no WinCE implementation for rtxSerial. Well, at least till 2002. Michal Hobot had written this port, but unfortunately, when you use the compiled binaries from page :

<http://mho.republika.pl/java/comm/>

and then you call for example something like this :

```
if (OBuffer.capacity() != 1)
    RS232Out.flush();
else {
    //System.out.println("!");
}
RS232Out.write(b);
RS232Out.flush();
```

you will get into trouble, because you will get following error :

```
java.lang.UnsatisfiedLinkError: nativeDrain
  at gnu.io.RTXPort.nativeDrain(Native Method)
  at gnu.io.RTXPort$SerialOutputStream.flush(Bytecode 47)
```

Now, Michal Hobot had send to the main maintenance of rtxSerial Trent Jarvi patch for it, but no one have cared to compile the sources again. Michal Hobot has not updated his library from his page, mentioned above, so you have to recompile the code once again. The problem is, that this sources were compiled and whole project was prepared for embedded tools 3.0 from Microsoft. This is not available already. So you have to do it with embedded visual c++.

There is also no kind of description on how to do it, so I decided to do it, for the case, I will need it some day again. This document describes it.

## 2. rtxSerial for WinCE

### 2.1. Getting WinCE rtxserial sources

Well that's pretty easy. Just click the link below, and the download starts :

<http://rxtx.qbang.org/pub/rxtx/rxtx-2.1-7r2.zip>

Now unpack the archive and under the folder WinCE there is a project from Michal Hobot with already patched file `gnu_io_RXTXPort.cpp`. That is the cause of the flush problem.

## 2.2. Getting embedded visual c++ and additional software

Embedded visual c++ 4.0 is free, so don't worry. You can get this from here :

<http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=24809>

Install this one in your favourite folder.

Then additionally you have to install at least SP3 for the embedded visual c++ 4.0. You can get this from here :

<http://www.microsoft.com/download/en/details.aspx?DisplayLang=en&id=4800>

Simply install this sp3 and that's it.

Now finally you have to install the **SDK for Windows Mobile 2003-based Pocket PC**. It's available from this page :

<http://www.microsoft.com/downloads/details.aspx?FamilyID=9996B314-0364-4623-9EDE-0B5FBB133652&amp;displaylang=e&displaylang=en>

Install this SDK.

## 2.3. Setting up rtxserial WinCE for build

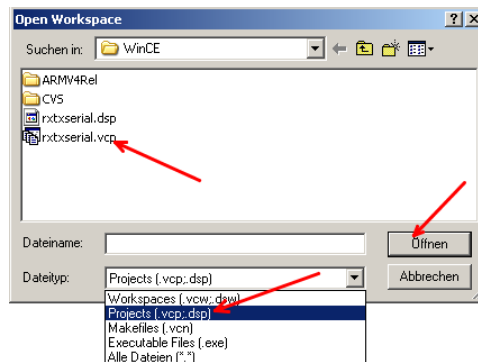
After successful instalation of the above software open embedded visual c++ 4.0. Now we have to set up some settings to build the dll with Michal Hobot's new patch (not so new, because from 2004, but we still need the dll's).

1. First off all open the workspace for the rtxserial project. Click **File** → **Open Workspace...** from the main menu :

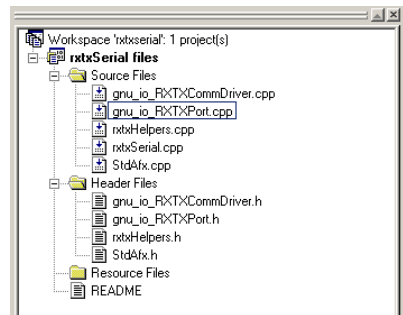
**Figure 1. Selecting workspace to open**



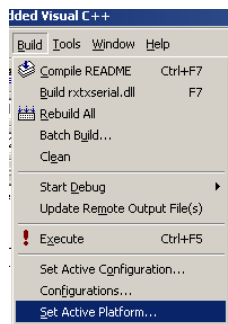
2. In the next dialog box navigate to the folder with unpacked WinCE folder of rtxserial library. Workspace for rtxserial is made with older tools, so you have to **filter** the files and search for **Projects(.vcp;.dsp)**. Then select the **rxtxserial.vcp** and click **OK** button.

**Figure 2. Opening rtxserial.vcp project**

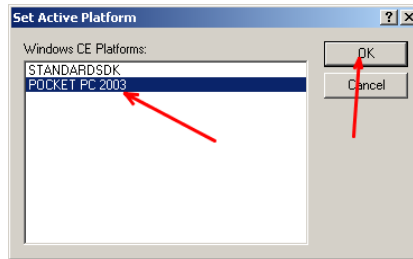
3. In the left navigation panel you should see opened project, something like this :

**Figure 3. Opened project rtxserial in navigation panel**

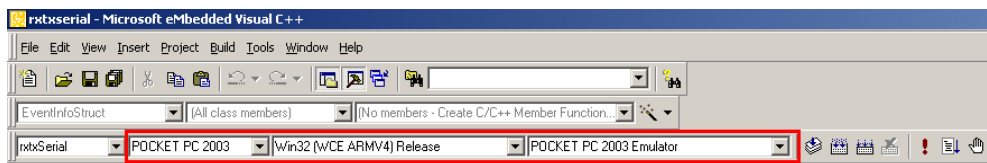
4. Now we have to change the default platform for the project. Select **Build** → **Set Active Platform...** from the main menu.

**Figure 4. Setting active platform menu item**

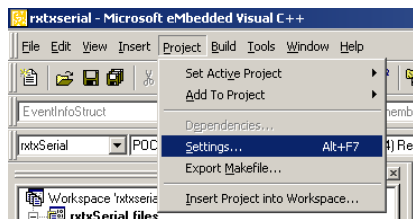
5. Another dialog box will open. After successful installing SDK for Pocket PC you should have additional platform to choose on this dialog box. So, choose **POCKET PC 2003**. And click **OK** button.

**Figure 5. Selecting POCKET PC 2003 as active platform for the project**

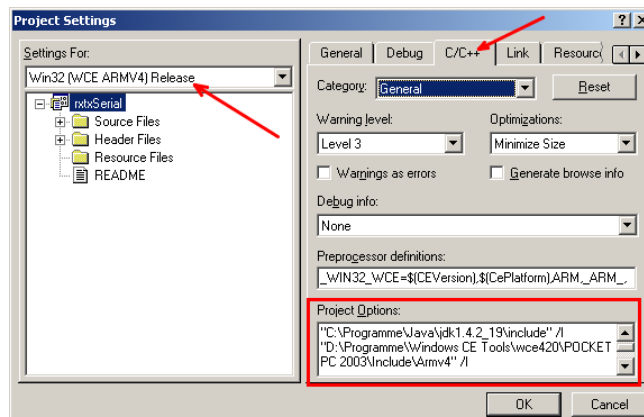
6. Once again check the settings in the main window for the project. Here you can choose between **Win32 (WinCE ARMV4) Release** and **Win32 (WinCE ARMV4) Debug**. I've chosen Release for I don't want to debug the build for now.

**Figure 6. Checking the correct configuration in main window of the project**

7. Now choose Project Settings... from the main menu. We have to add some additional parameters for the compiler, so that he will know where to find the appropriate include files.

**Figure 7. Selecting Settings menu item of the project**

8. Now in the C/C++ tab you have to add in the field Project Options at the end of the entry path's to the include files:

**Figure 8. Setting up compiler options**

So, this is really important. You have to add the path to include files of java (you are building a dll that is using JNI interface for accessing native methods from Java language). And it is pretty important too, that you

should remember that on the windows mobile devices, mobile version of JDK is 1.1. There is no Java 1.5, or 1.6 for mobile devices yet. (in my work I use IBM JVM, but there are plenty of other JVMs available for mobile devices). And you have to remember that the Java application that you are building for the device should be compiled AT MOST with Java 1.4 JDK on the desktop PC. So the path should indicate the 1.4 JDK include directories. So, here in this Project Options field add at the end of the field following lines for Java path :

```
"C:\Programme\Java\jdk1.4.2_19\include" /I
"C:\Programme\Java\jdk1.4.2_19\include\win32" /D
```

And additionally you have to add path to Pocket PC includes. In my configuration of Pocket PC it is so (you have to find your directory though) :

```
"D:\Programme\Windows CE Tools\wce420\POCKET PC 2003\Include\Armv4" /I
```

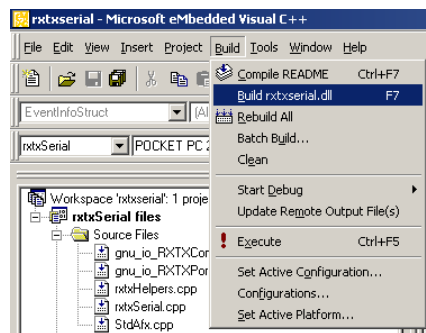
Now click the OK button on this dialog box.

## 2.4. Building dll

Right, now we are ready to build rtxserial.dll. So, let's do the following :

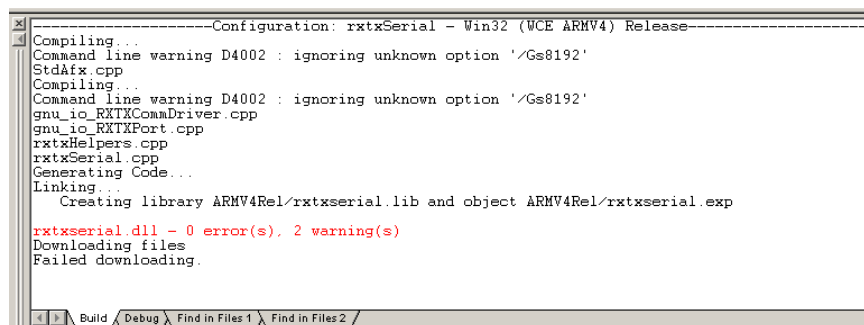
1. Select Build Build rtxserial.dll from the main menu

**Figure 9. Selecting Build item from main menu**



2. Now, if in the message window will no error problem appear, then this means you have your rtxserial.dll build.

**Figure 10. Build process completed**



Isn't it great?

Your fresh dll should be under the folder <WORKSPACE>\WinCE\ARMV4Rel. Now all you have to do is to copy this to the virtual machines bin folder on the mobile device. And you can talk to the serial ports on the mobile device through Java.

The compiled dll and the sources with corresponding additionally added jars for Java communication you can find under <http://www.mediafire.com/?683vcb8v7kwtt9h>