



# Mobile Phone Controlled PODdevices™ A Tutorial Part 2 - PODapp™ Mobile Phone application

by Matthew Ford 5<sup>st</sup> December 2010

© Forward Computing and Control Pty. Ltd. NSW Australia

All rights reserved.

## PODapp™ – the Mobile Phone Application (Protocol for Operations Discovery)

### Outline of the Tutorial

#### Part 1 – Introduction to PODdevices™

- [Introduction - Why PODdevices?](#)
- [What can PODdevices™ do.](#)
- [Protocol for Operations Discovery \(POD\).](#)

#### Part 2 – POD Mobile Phone Application

- [PODapp - Choosing your Mobile Phone](#)
- [PODapp - Setting up the Connections](#)
- [Connecting to your PC](#)
- [Installing the PC Terminal Emulator](#)
- [Connecting PODapp™ to TeraTerm](#)

## Part 3 – Led Controller PODdevice™ Messages

- [The PODdevice™ Main Menu](#)
- [The Fade On / Fade Off Command](#)
- [The Up and Down Commands](#)
- [The Sub-Menu](#)
- [The Set Level Sub-Menu](#)
- [Low, Medium, High and Off Commands](#)
- [The Data Sub-Menu](#)
- [The Light Level Command](#)
- [The Led Current Command](#)

## Part 4 – Coding the Led Controller PODdevice

- [Coding the Led Controller PODdevice](#)
- [The PODdevice™ Message Handler](#)
  - [PROCESS\\_RECEIVED\\_CHAR](#)
  - [PROCESS\\_CMD\\_CHAR](#)
  - [Menu Commands](#)
  - [MAIN\\_LOOP](#)
  - [BRANCH\\_IF\\_TORCH\\_OFF\\_OR\\_FADING\\_DOWN macro](#)
  - [Level Setting Commands](#)
  - [On/Off and FadeOn/FadeOff Commands](#)
- [Conclusion](#)

[Part 1](#) of the tutorial introduced PODdevices™ and the covered the specification of the operation discovery protocol. This part of the tutorial covers a reference implementation of a J2ME mobile phone Java application the supports the protocol. This application can be used to control any PODdevice™ that has a bluetooth/serial connection. This is what makes this mobile phone application so useful. It is the PODdevice™ that tells the phone what functionality is available. The phone then just displays the commands and lets you choose.

## PODapp™ – Choosing your Mobile Phone

To use the PODapp, you need a suitable mobile phone. Your Mobile Phone needs that supports the Java JSR-82 Java Bluetooth API. My old phone did not so I purchased an inexpensive Samsung C3110. I used the website [http://www.club-java.com/TastePhone/J2ME/MIDP\\_Benchmark.jsp](http://www.club-java.com/TastePhone/J2ME/MIDP_Benchmark.jsp) to check if the phone would support Java JSR-82. Or you can just download the PODapp™ and see if it runs.

The [PODapp.jar](#) (updated 14<sup>th</sup> Feb 2011 to V1.1.0) and [PODapp.jad](#) files are available for download. Right click on the links and choose Save Link As.. and save both these files to your hard drive and then transfer them to your phone. (See [Loading Java Programs to your Phone](#))

When you run the program you should see a display something like this.



## PODapp – Setting up the Connections

Once you have confirmed the PODapp™ runs on your phone, you need to set your own connection list. This is done by editing the PODapp.jad text file. First take a copy of the file in case your changes don't work.



The last line is the only line you need to change

**PODdevices: 001E3DEDBC88~PC,0|00066601E6A0~POD Led Controller**

The **PODdevices:** line is where you define the bluetooth addresses and display names and timeouts for each

PODdevice you want to connect to. The format is

```
<bluetoothAddress> [[ ~ <displayName> ] [ , <charTimeOut>]] [ | <bluetoothAddress> [[ ~
<displayName> ] [ , <charTimeOut>]] ]*
```

The [ ] indicate optional sections, [ ]\* means zero or more of these sections.

That is, one PODdevice™ connection must be defined and you can add as many others as you like each separated by the | character. Within each connection the display name and timeout are optional. If the displayName is omitted then the bluetooth address is displayed in the Connect To menu. If the character timeout is omitted, then the default of 10mS is used. Setting a character timeout of 0 means the connection never times out. This is useful for testing.

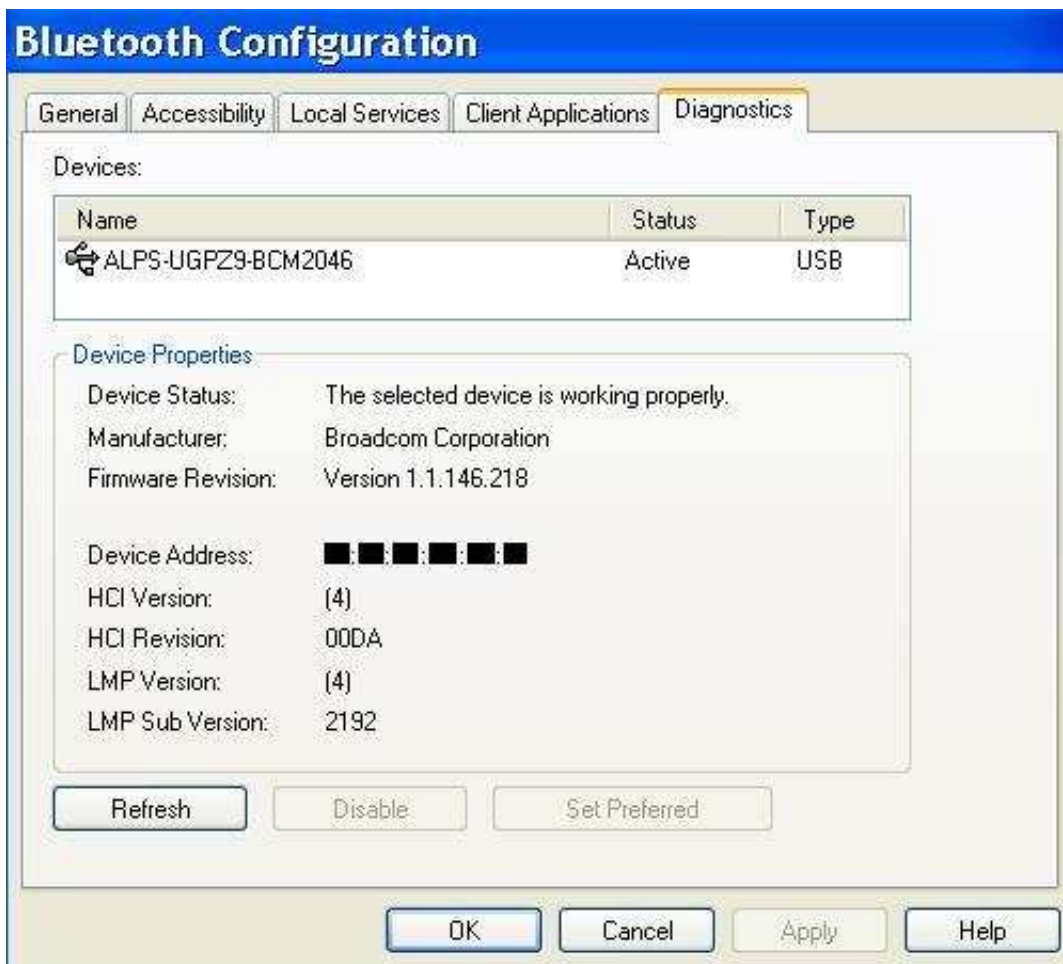
Looking at the current settings,

**PODdevices: 002F3DFDBC77~PC,0|00176521F6A1~POD Led Controller**

This line defines two PODdevice™ connections separated by |

For the first connection the bluetooth address is **002F3DFDBC77** , you should change this to the bluetooth address of your laptop. Look in the Control Panel under Bluetooth Connection for your Device Address address. The screen shots below are from my Sony Vaio running Windows 2000. Your Bluetooth configuration screen may look different.

Don't enter the :s just put in the 12 characters, shown below blanked out.



character timeout, separated by , in this case **0** which means the connection never times out waiting for the start of message or for the next character in a message.

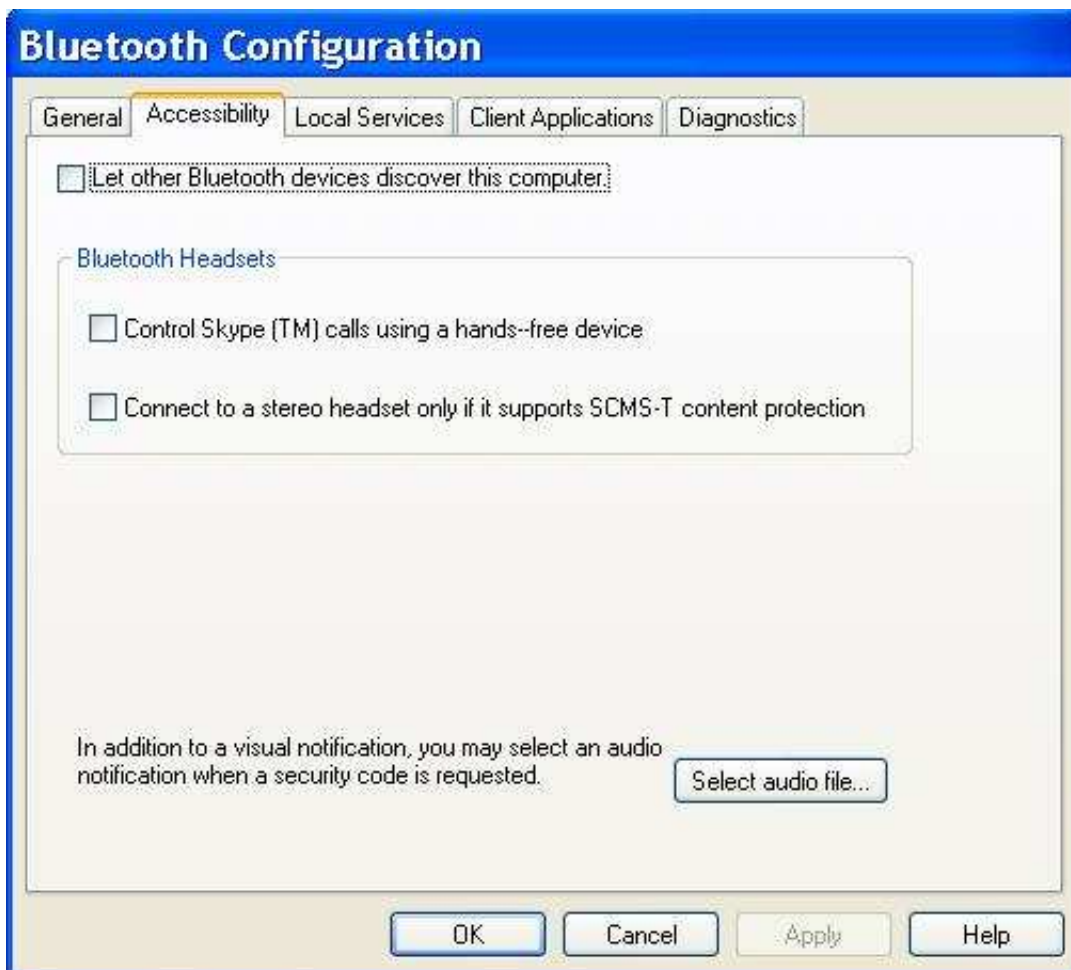
The second connection is separated from the first by | and defines the bluetooth address of the **POD Led Controller**. You can read the bluetooth address off the label on the SparkFun's bluetooth module. This connection does not define a character timeout so the default of 10mS is used. This means the POD Led Controller must start to respond to a command within  $5 \times 10\text{mS} = 50\text{mS}$  and must send each message character within 10mS of sending the last one, otherwise the PODapp™ will assume the connection is broken and will disconnect.

The second last line is the **PODdebug:** setting. If this setting starts with **y** or **Y** then the debug mode is turned on and the PODapp™ will add a Msgs button item which will display all the messages to and from the PODdevices. When you have finished your debugging either delete this line or just change it back to **PODdebug: no**

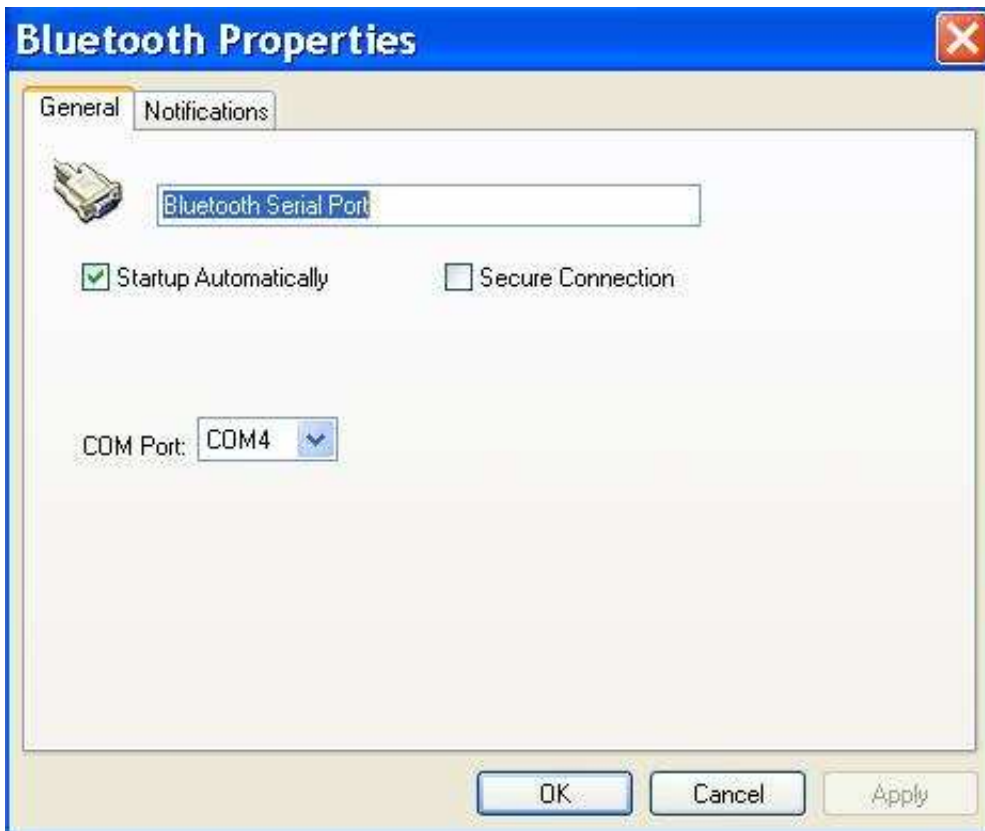
## Connecting to Your PC

Once you have edited the PODapp.jad file with your PC's bluetooth address, you can transfer the edited file to your phone and connect to your PC to test the PODapp. This lets you compose the messages your PODdevice™ will send to advertise its functionality and see what responses the PODapp™ will send when the user selects a menu item.

To connect to your PC, you can and should leave your PC bluetooth set to NOT be discoverable. This is your primary security protection. With this option un-ticked only devices that know your bluetooth address can request a connection.



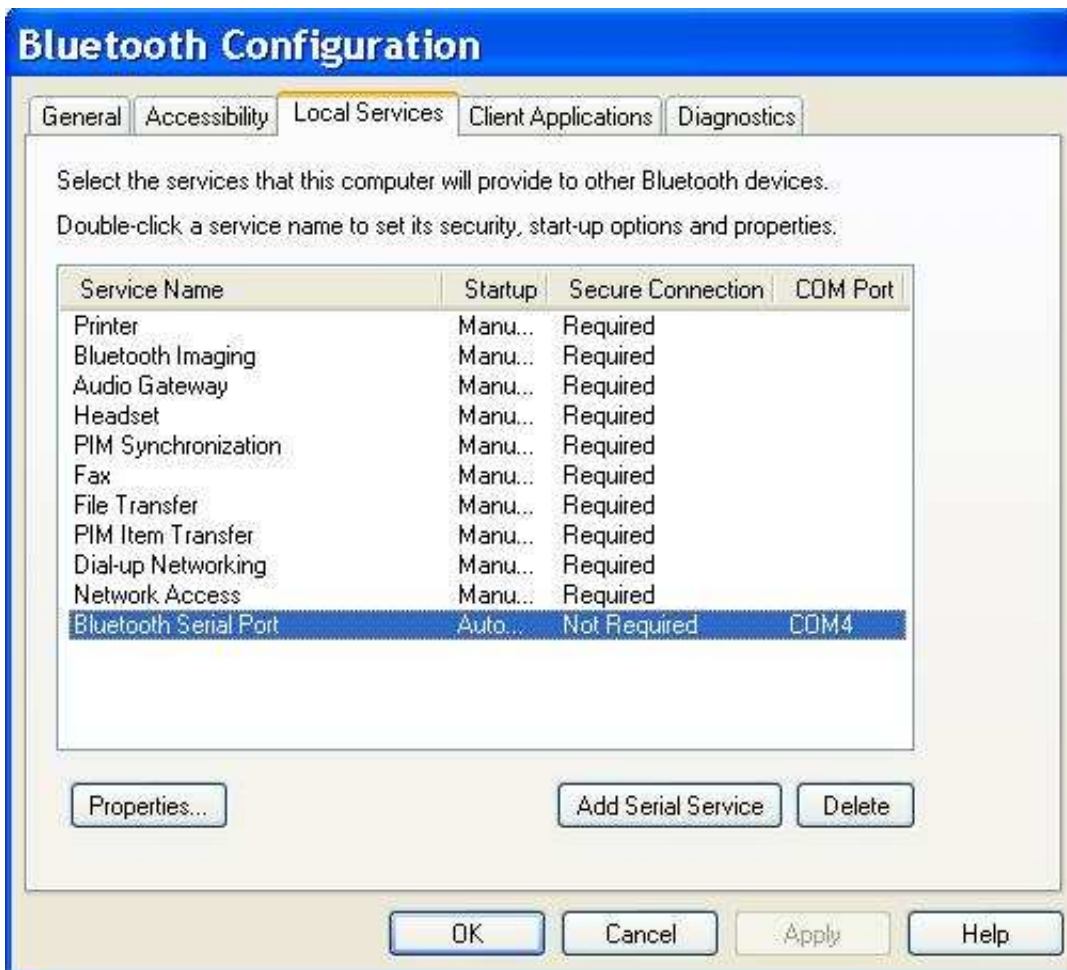
Next setup your Serial Bluetooth Connection. In my case this dialog is accessed from the LocalServices tab.



Set it to start up Automatically but do NOT secure it. This free version of PODapp™ does not support secure connections. Security is provided by having to know the exact address of the PODdevice™ in order to connect to it.

Clicking OK on the above dialog updates the Local Services Dialog below.





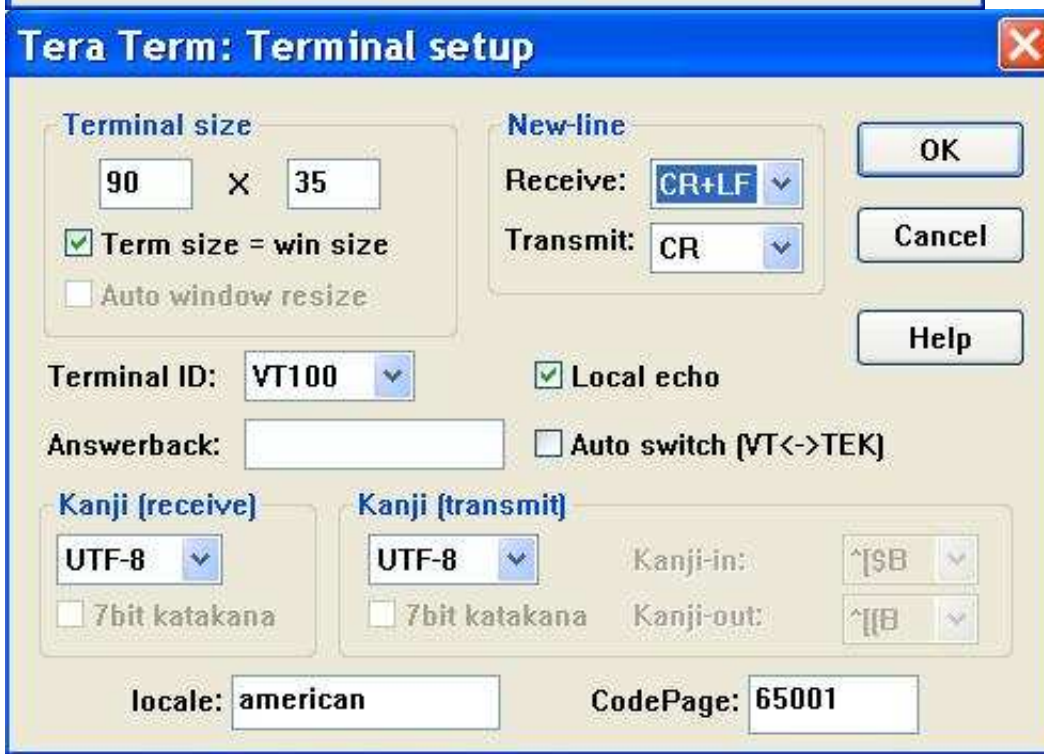
## Installing the PC Terminal Emulator

To communicate with the PODapp™ via the bluetooth serial connection on your PC, you need a terminal emulator. I use TeraTerm (<http://ttssh2.sourceforge.jp>) but there are lots of other possibilities, including Hyperterminal

Open TeraTerm and start a new connection. Choose the

COM port that matches your Bluetooth Serial Port.





Open the Setup → Terminal dialog and tick Local Echo and set New-Line Receive: to CR+LF. Local echo lets you see what you are typing and Receive CR+LF makes the display more readable.

## Connecting PODapp™ to TeraTerm

You need to start TeraTerm before connecting from your mobile phone PODapp, because the PODapp™ sends the getMainMenu message, {,}, immediately after the connection is established. If TeraTerm is not already running and listening to the bluetooth COM port, then you will miss this message.

Start the PODapp™ on your mobile phone and connect to your PC.



The TeraTerm window shows the {.} message from the PODapp.



The PODapp™ expects the PODdevice™ to reply with its main menu of functionality. Type in the Main Menu response. Actually it is easier to type up the command in a text editor and then cut and paste it to the TeraTerm window because TeraTerm sends every character you type, immediately, including your typos.



The PODapp™ will display



The next sections of this tutorial will cover defining the functionality of the POD Led Controller and coding it into microcontroller.

---

PODdevice™ and PODapp™ are trade marks of Forward Computing and Control Pty. Ltd.

---



Contact Forward Computing and Control by [email](#)

©Copyright 1996-2012 Forward Computing and Control Pty. Ltd. ACN 003 669 994