



Mobile Phone Controlled

PODdevices™ V1.1

A Tutorial Part 1 - Introduction to PODdevices™

by Matthew Ford 2nd December 2010

© Forward Computing and Control Pty. Ltd. NSW Australia

All rights reserved.

Introduction to Mobile Phone control of PODdevices™ V1.1 (Protocol for Operations Discovery)

This tutorial extends the [Bluetooth Controlled Led Driver](#). It provides a complete solution for controlling arbitrary microcontroller devices from your mobile phone. The solution presented here consists of a Mobile Phone J2ME java application, a protocol specification and a Led Driver that implements the protocol.

Outline of the Tutorial

Part 1 – Introduction to PODdevices™

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

Part 2 – PODapp™ - 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](#)

Why PODdevices?

I found the bluetooth connectivity to the Mobile Phone Controlled Led Driver very useful and wanted to extend it to my other torches, led lighting and household devices. But I did not want to have to remember what each device could do and did not want to have to re-code the mobile phone application to support each new device. The solution I come up with was to define a simple operations discovery protocol by which the mobile phone application asked the device what commands it supported and then let the user choose one.

PODdevices™ are devices that support this protocol, as defined in [PODSpecificationV1.1.pdf](#)

In this tutorial I will provide the mobile phone J2ME java application and the code for the converting the Mobile Phone Controlled Led Drive into a PODdevice. If you don't make the Led Controller, you can play around with the mobile phone application using the bluetooth connection on your laptop.

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

What can PODdevices™ do?

Before going into the details, let's see what a PODdevice™ can do, in this case the POD Led Controller.

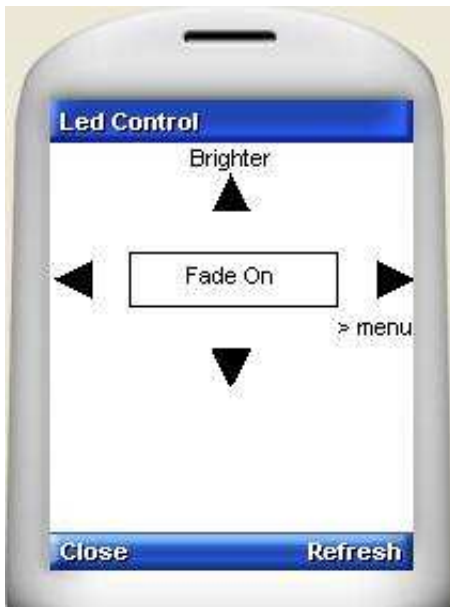
Starting up the POD mobile phone application shows the Connect To screen



This lists the names of the Bluetooth devices you can connect to. These devices (and their Bluetooth addresses) are configured by editing the PODdevice.jad file.

This mobile application can control any device which has a bluetooth/serial connection that implements POD (Protocol for Operations Discovery). The protocol itself does not mandate using bluetooth, other implementations could use wifi, internet or radio to make the connection and transfer the messages.

Choosing "Connect", connects to the selected PODdevice™ and asks it for its main menu. The PODdevice responds a Navigation Menu e.g.



Pressing the phone's corresponding home navigation key sends the command associated with the ON label to the PODdevice, in this case the POD Led Controller. The POD Led Controller turns the Led on and replies with a navigation update



Pressing the right direction key sends the command associated with the Menu label to the POD Led Controller. The POD Led Controller sends back "Settings Menu e.g.



Sending the “Set Level” command, the POD Led Controller responds with a new sub-menu which lets you set the Led level to High, Med, Low or Off

Sending the “Data” command, the POD Led Controller responds with a sub-menu of



Sending the “Light Level” command, the POD Led Controller responds with the Streaming Raw Data message and then sends the current light level details in free text.



Other possible messages a PODdevice™ could respond with include requesting text input, numeric input and single and multi-selections from a list.

What makes this mobile phone application so useful is that the same application shown here can control many different devices. It is the device that tells the phone what functionality is available. The phone then just displays the commands and lets you choose. Languages other than English can also be supported.

Protocol for Operations Discovery (POD)

The specification for POD is detailed in [PODSpecificationV1.1.pdf](#). It defines the message formats and some minimal connection details. The specification does not define how the messages are displayed to the user.

Here is an example of the initial connection message exchange. Each message, except streaming raw data, is contained within a start and end character { } . When the mobile phone application establishes a connection, it first sends the getMainMenu message, { . } , to the PODdevice. The PODdevice™ responds with a menu of functionality that it supports. In this example the PODdevice responds with { . | **o~on** | **u~up** | **d~down** | **s~set Level-** **>|a~data->** } . This message defines a menu which defines 5 items *on*, *up*, *down*, *set Level->* and *data->*



Each of these menu items is associated with a command that the PODdevice will respond to. For example when the user selects the *on* menu item, the application will send the associated command **o** to the PODdevice™ (as the message {**o**}).

The PODdevice™ then executes that command and responds to the application with some message, perhaps an empty one {}, to indicate the PODdevice™ received the command and has executed it. In this case the POD Led Controller responds with the message {**o~off**} This updates the display text for the **o** command to show *off*.



For a detailed description of the messages, examples and a sample displays, see the [PODSpecificationV1.1.pdf](#).

For simple devices, single letter commands are sufficient and no processing of user input is required. This minimizes the coding required in the microcontroller to implement the protocol. See the POD Led Controller section ([Part 4](#)) for a detailed code example.

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