

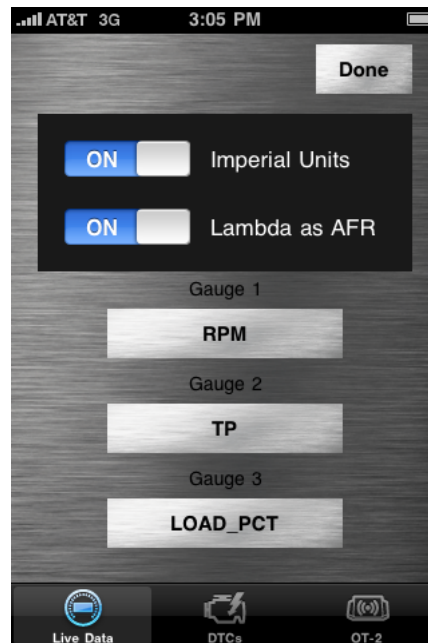
## Disclaimer

This document, as well as the accompanying source code, is presented “**as is**”, without warrantee or guarantee. If it does not work, or you blow something up, the developers and Innovate Motorsports **are not responsible**. The source code is provided without restriction. However, some symbols and logos in the provided artwork are protected property of Innovate Motorsports Inc. Those images may not be used or redistributed without the express consent of Innovate Motorsports Inc.

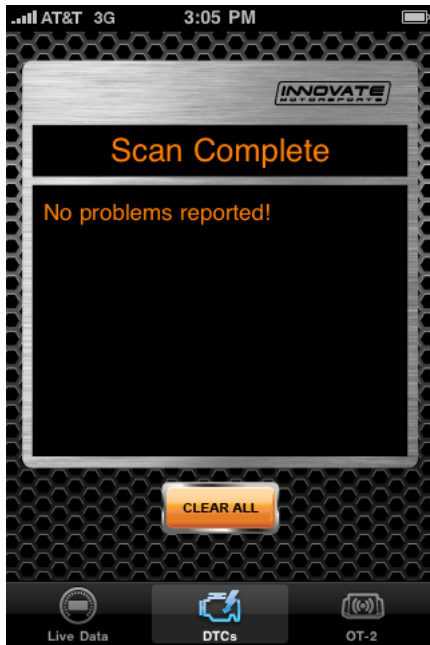
## OT Mobile

V1.0

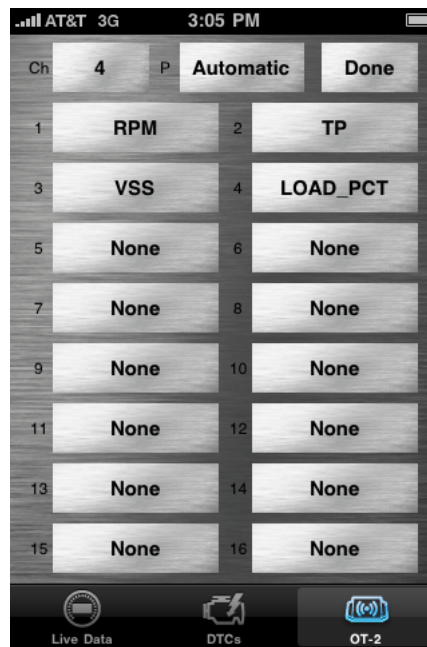
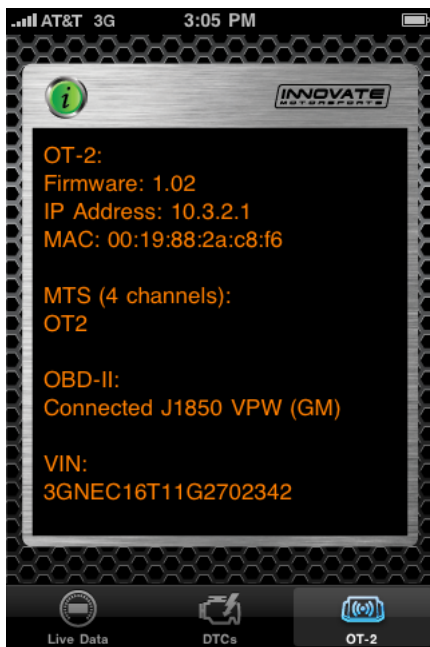
This is a relatively simple, but complete, iPhone application that accesses the Innovate Motorsports OT-2 via Wi-Fi.



It has a simple three gauge display, which can be customized by pressing the info button/connection status indicator to show any three channels (either from the OT-2 or any attached MTS compatible devices (ex. LC-1, LM-2, etc.)



There is a pane to view and clear Diagnostic Trouble Codes.



And a pane that displays some basic information about the OT-2 and vehicle (if connected). Pressing the info button/connection status indicator on this pane let's you configure the OT-2, ala LM Programmer.

## Required Tools

To build and run this application on the iPhone simulator, you will need to download the current version of the iPhone SDK, which includes the XCode development environment.

The SDK only runs on Macintosh, not Windows (of course).

If you wish to run this application on an actual device, you will need to join the iPhone Developer Program. This is because building for devices requires code signing and you need to join the program to get a certificate.

## Building

1. Load the project file with XCode ("OT Mobile.xcodeproj")
2. Make sure that "Simulator 3.0 Debug" is selected as the target
3. Click "Build and Debug"

The application should build and run in the launched simulator. If your Mac has an "Airport" (Wi-Fi adapter) and you connect it to the ad hoc network published by an OT-2, the simulated application will find and connect to the hardware.

## Notes

First, what this application is not. This is not a tutorial on Objective C or iPhone SDK development. Quite the opposite, C, Objective C, and C++ are pretty freely intermixed. What the application provides is a working example of most OT-2 SDK functionality on the iPhone platform.

The basic model is a UI, and an OT-2 engine. The engine runs in it's own thread and is autonomous. That is, once you start it, it should maintain and restore OT-2 connectivity until you expressly stop it.

The two sides (UI and engine) are asynchronous. The OT-2 engine generates notifications, which you can monitor (like 'connection status change' or 'new MTS packet received'). When the UI wishes to do something, it makes a command

request, which the OT-2 engine eventually acts on and responds to, again by notification.

Networking was done entirely with the underlying BSD sockets layer, not the higher level wrapper layers that Apple provides. If you are not running in a separate thread, those wrappers, which are run loop aware, are probably a good idea. However, you will still want to use the TCP\_NODELAY option on your underlying TCP socket (see the source). Generally, you can set this option, then use the native socket when instantiating one of the higher level interfaces.

Questions and comments can be directed to [support@innovatemotorsports.com](mailto:support@innovatemotorsports.com).