

---

## **Bluetopia® MSP430 MFi Information**

Introduction .....	1
Overlay.....	1
C/C++ Application Development Notes.....	2
CCS/IAR Toolchain Library Notes .....	2
Building the Sample Applications.....	2
Sample Application Notes .....	2
iSPPDemo .....	3

### **Introduction**

Thank you for selecting the Bluetopia, the Bluetooth Protocol Stack by Stonestreet One. This release has been prepared to support the MSP430 Platform using either the CCS or IAR MSP430 tool chain. Please Contact Stonestreet One for any technical support issues.

This release includes the MFi overlay for the generic TI MSP430 released maintained by Stonestreet One.

### **Overlay**

This release only contains the files necessary to add MFi support to the generic TI MSP430 release that is maintained by Stonestreet One. All other necessary files must be obtained from Texas Instruments websites after following a click through license agreement. The URL for the latest provided software is below:

<http://www.ti.com/tool/stonestreetone-bt-sdk>

## C/C++ Application Development Notes

This section details specific usage information on how to effectively use Bluetopia with C/C++ Compilers.

In order to include all necessary prototypes/constants, the programmer needs to only include the **SS1BTPS.h** header file (for the core library). In order to include all necessary prototypes/constants for profiles, the programmer needs to only include the **SS1BTxxx.h** for the specific profile (where, 'xxx' is the profile name).

Users using C++ **MUST** use the following code to include the **SS1BTPS.H** Header File in their C++ Source:

```
extern "C" {  
    #include "SS1BTPS.h"  
}
```

Users using C (and NOT C++) do NOT need to include the above code snippet and can simply include the **SS1BTPS.h** Header File as normal, for example:

```
#include "SS1BTPS.h"
```

All API structures that are used with Bluetopia at the hardware device interface level are aligned on default packing boundaries. This shouldn't pose much of a problem because most of the structures are aligned such that they would not cause too many packing problems otherwise. Lower level structures are aligned based on compiler switches as specified by the customer.

## CCS/IAR Toolchain Library Notes

The libraries contained in this distribution were compiled using either CCS or IAR MSP430 tool chain. Users simply are required to link with the **libBluetopia.a** library for core stack functionality (stored in the folder **lib\CCS** or **lib\IAR**). All other profiles and support libraries have a corresponding library directory (also named **lib\CCS** or **lib\IAR**) which contains the library required for linking.

A CCS and IAR Project is provided for the sample applications (source is included).

## Building the Sample Applications

This release is intended to be built with either Code Composer Studio IDE or IAR's Embedded Workbench for MSP430. A CCS Project is included in the \Project\CCS folder and an IAR Project is included in the \Project\IAR folder.

## Sample Application Notes

This release includes a sample application directory with source code which demonstrates the use of the Bluetopia stack regarding the provided samples is described below.

Input and Output can be utilized by plugging in the MSP430F5438A Experimenter board Serial to USB debugging connector to a PC. The ez430-RF256x when connected to the USB stick can also communicate with a PC. The parameters for this port are:

Baud:	9600
Data Bits:	8
Parity:	None
Stop Bits:	1
Flow Control:	None
Local Echo:	Off

### **iSPPDemo**

This application also shows how to utilize the iSPP/MFi module and how to handle the different iSPP callback events. This will act as either a Server or Client.