Whistle Labs Bluetopia (MQX)
Release Notes
Release 4.0.1.3

Stonestreet One
9960 Corporate Campus Drive
Suite 3300
Louisville, KY 40223

Revision History

Rev	Updated Areas	Date	Author
4.0.1.1	First Release	1/28/2013	M. Funk
4.0.1.2	Implemented iAP Transport	3/1/2013	S. Hishmeh
4.0.1.3	Added libs built for CW	4/19/2013	M. Funk

1. Files Included in this Release

1.1 Release files

All source and libraries are provided in a Zip file format. The archive name is 'WhistleLabs_Bluetopia_4_0_1_3_K60_MQX.zip'.

1.2 Documentation

Various Documents are provided in PDF form. These documents include:

- Bluetopia API documentation
    - Core Protocol Stack
    - Generic Access Profile Service
    - Generic Attribute Profile
    - iAP Serial Port Profile

- Bluetopia Porting documentation
    - Architecture Overview
    - System Call Requirements
    - BTPSKRNL (O/S abstraction layer)
    - HCITRANS (HCI Transport abstraction layer)

All documentation is located under the:

Documentation

subdirectory.

1.3 Libraries/Source Code

All libraries and source code for the Bluetooth protocol stack is located under the:

Bluetopia

subdirectory.

All sample source code is located under the:

TWR-K60N512

subdirectory.

2. Release Notes

2.1 Overview

As previously discussed, this release includes support for the Freescale K60N5612 processor. The precompiled libraries shipped with this release were built using the RealView compiler. The pre-compiled binary library was compiled with the C compiler (not with C++ compiler). The name of the libraries is:

- **Bluetopia.a** – Core Bluetopia library (HCI, L2CAP, SDP, RFCOMM, SPP, GAP)

This release also contains source files that have been ported to MQX on the K60. These files are located in their own directories are named:

- **BTPSKRNL.c** – O/S abstraction layer
- **HCITRANS.c** – HCI UART transport abstraction layer
- **BTPSVEND.c** – Bluetooth chipset vendor specific initialization abstraction layer.

All locations in the above files have the following C++ comment where porting code needs to be written:

//xxx

Porting documentation for the required functions is included for reference.

Finally, this release contains a sample applications in source form. This application is a fully functional application that shows the following usage:

- Initializing/shutting down of the stack
- Device Discovery (Inquiry and SDP)
- Device Pairing
- Serial Port Profile (SPP) usage
- Local Device Management (class of device, local name, device address, connectability modes, discoverability modes, pairability modes, etc).

The sample application provides a command line interface (through a UART or some other mechanism) that can be used to issue commands in real-time and watch the results. These samples should easily be modifiable to add/change features during testing/development.

2.2 Build Notes

This release has been ported to MQX and the TI CC2564 baseband controller. Please see the next section about porting notes.

2.3 General Porting Notes

Building a port of Bluetopia for a new platform consists of the following steps. Information for the various steps will further be described in its own section below:

- Port BTPSRKNL.c to support the target platform (provided)
- Port HCITRANS.c to support the target platform (provided)
- Port BTPSVEND.c to support the target Bluetooth chipset (provided)
- Create the sample/test application
- Create a project that includes the above ported files, the sample/test application, and links with supplied binary libraries.

2.4 BTPSKRNL Porting Notes

For information on porting BTPSKRNL.c to the target platform, please see the attached BTPSKRNL porting documentation. All functions that need to be replaced/implemented contain the following characters (in a comment):

xxx

2.5 HCITRANS Porting Notes

For information on porting HCITRANS.c to the target platform, please see the attached HCITRANS porting documentation. The HCITRANS.c file located in the Core directory should be used (Multi-threaded version). All functions that need to be replaced/implemented contain the following characters (in a comment):

xxx

2.6 BTPSVEND Porting Notes

As previously mentioned, this release ships with a sample Vendor Specific implementation that supports the TI CC2564 chipset. This module can be changed to support other chipsets if required. Stonestreet One can provide guidance in the configuration/implementation of this module for a specific target platform/chipset. This module is primarily to configure the specific Bluetooth chipset. This typically entails:

- Required chipset Patch RAM
- Required chipset default Bluetooth values (Name, Class of Device, etc)
- Radio Tuning parameters (specific to physical board layout)
- Means to apply a unique BD_ADDR to the chipset

The contents of what is actually downloaded to the chip is typically provided by the Bluetooth chipset vendor. The implementation of how to download these configuration parameters is what is contained in this module.

2.7 Sample Application Notes

The sample applications provided demonstrate how to utilize the various Bluetopia Application Programming Interfaces provided for each profile (GAP, SPP, GATT, GAPS, and ISPP). The sample applications can be found under the following subdirectory:

Sample

SPPDemo
SPPLEDemo
ISPPDemo

The above mentioned sample should serve as a good starting point because each Bluetooth task is broken up into a discreet function. The sample also provides the asynchronous event handlers and, more importantly, the code that is required to be executed in response to the asynchronous events. Note that the sample above provides a fully Bluetooth SIG qualifiable application for the above mentioned profiles/protocols.