Using the Flexible Build Library for STM32

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

This document is the proprietary property of Stonestreet One, Inc., Louisville, Kentucky, USA and is considered CONFIDENTIAL.

Revision Information

This is version 1 of this document, last updated 15 June 2015.

Rev	Updated Areas	Date	Author
1	Initial Release	1/1/2014	M. Funk
1.1	Update from SSO	1/1/2015	SSO
1.2	Update for STM32	25/1/2015	Doron Keren
1.3	Update with processor type	15/6/2015	Doron Keren

1 Introduction

The Flexible Build Library (FBL) is a mechanism that allows the user to generate different versions of the Bluetopia library tailored to the specified use case. This allows the user to reduce the flash memory requirements of the Bluetopia stack by removing code that is not needed by the user's specific application. Some examples are adding or removing Low Energy support, adding or removing SPP support and adding or removing SCO audio support.

2 Prerequisites

2.1 Perl

The provided FBL scripts require that Perl has been installed. Perl for Windows can be obtained free of charge from ActiveState. The Perl build scripts which are located under "Bluetopia\Build\NoOS or FreeRTOS" in the release can be run from the Windows command line after Perl has been installed.

2.2 IAR or Keil STM32 Build Tools

For IAR the FBL script "fbl_iar.pl" requires that the IAR for ARM archiver utility IAR for ARM (iarchive) must be in the %PATH% variable.

For Keil uVision the FBL script "fbl_rvmdk.pl" requires that the Keil ARM archiver utility, RVMDK Archiver (armar), must be in %PATH% variable.

3 Using the FBL

Once the prerequisites enumerated in Section 2 have been satisfied the FBL is ready to be used. The FBL is run on the command line and will build new libraries based on the provided options. The location of the FBL files and the script options are provided below.

3.1 File Locations

All FBL files are described below:

- Bluetopia\Objects — Root directory for all FBL related Object files
    - fbl_rvmdk.pl — FBL build script for Keil.
    - fbl_iar.pl — FBL build script for IAR.
    - KEIL — Directory that contains KEIL objects files used by fbl_rvmdk. pl script.
    - IAR — Directory that contains IAR objects files used by fbl_iar.pl script.

3.2 FBL Script Options

The following FBL script options and argument are valid for both the KEIL and IAR FBL scripts. All the Bluetooth profiles options are optional and if not specified then the resultant library(ies) will not contain support for that feature. The processor type argument is mandatory and the script will not run without processor type argument. Note that SDP server support is required and will always be included in the resultant core library.

Processor type argument:

- 16_M3 — For Arm Cortex M3 with wchar=16bit.
- 16_M4 — For Arm Cortex M4 with wchar=16bit.
- 16_M4.fp_HW_FP — For Arm Cortex M4 with wchar=16bit and HW floating point.
- 32_M3 — For Arm Cortex M3 with wchar=32bit.
- 32_M4 — For Arm Cortex M4 with wchar=32bit.
- 32_M4.fp_HW_FP — For Arm Cortex M4 with wchar=32bit and HW floating point.

Bluetooth Profile options:

- --sppserver — If specified resultant core library will contain SPP Server support.
- --sppclient — If specified resultant core library will contain SPP Client support.
- --sdpclient — If specified resultant core library will contain SDP Client support.
- --lemaster — If specified resultant core library will contain LE Master support.
- --leslave — If specified resultant core library will contain LE Slave support.
- --sco — If specified resultant core library will contain SCO Audio support.

- --gattserver If specified resultant GATT library will contain GATT Server support.
- --gattclient If specified resultant GATT library will contain GATT Client support.
- --hidhost If specified resultant HID library will contain HID Host support.
- --hiddevice If specified resultant HID library will contain HID Device support.
- --largemtu If specified resultant Core library will contain support for the highest possible L2CAP MTU (required for Audio support).

3.3 Resultant Files

The FBL can generate up to three separate libraries based on the options, detailed in Section 3.2, which are specified as the arguments to the FBL script. The libraries that may be generated are described below.

- Core Library This is the main stack library. Every application that uses the Bluetopia stack must link with this library in order to have Bluetooth functionality.
  - libBluetopia_16_M3.a If no LE support is specified when using the FBL script (i.e. "--lemaster" and "--leslave" are not specified and _16_M3 as the processor type) then the resultant core library is called "libBluetopia_16_M3.a". The library is generated in the path- \Bluetopia\lib_fbl\generated_libs\lib\rvmdk\FreeRTOS.
  - libBluetopia_LE_16_M4.a If LE support is specified when using the FBL script (i.e. either "--lemaster" or "--leslave" is specified and _16_M4 as the processor type) then the resultant core library is called "libBluetopia_LE_16_M4.a" to denote that LE support is included. The library is generated in the path- \Bluetopia\lib_fbl\generated_libs\lib\rvmdk\FreeRTOS.

- GATT Library This library provides the support for the GATT profile. Every application that needs to use GATT must link with this library. In order for GATT to work the core library that is also linked with the application must have LE support (i.e. the core library linked with the application is called "libBluetopia_LE_16_M4.a" with M4 and wchar=16bit as the processor type).
  - libSS1BTGAT_16_M4.a This is the name of the GATT library. This library is only generated by the FBL script if either "--gattserver" or "--gattclient" is specified and _16_M4, when using the FBL script.

- HID Library This library provides support for the traditional HID profile (HID over BR/EDR). Every application needing HID support must link this library. This library can be linked with either versions of the core libraries ("libBluetopia_32_M4.a" or "libBluetopia_LE_32_M4.a", when the processor type is M4 with wchar=32bit).

- libSS1BTHID_32_M4.a This is the name of the HID over BR/EDR library. This library is only generated by the FBL script if either "--hidhost" or "--hiddevice" is specified and _32_M4, when using the FBL script.

3.4 Examples

Some examples of using the FBL scripts and what the files generated by the FBL script are shown below. For this document we will show the usage for the KEIL script. However the usage of the IAR script is exactly the same.

Generating a core library with Processor type M4 wchar=16, SCO support, SDP server and client support, and SPP server and client support:

Usage: fbl_rvmdk.pl 32_M4 --sco --sdpserver --sdpclient --sppserver --sppclient

Generates: libBluetopia_32_M4.a

Generating a core library with Processor type M4 wchar=16, SDP server and SPP server support. Also generating a HID library with device role support:

Usage: fbl_ rvmdk.pl 16_M4 --sdpserver --sppserver --hiddevice

Generates: libBluetopia_16_M4.a, libSS1BTHID_16_M4.a

Generating a core library with Processor type M3 wchar=32, SDP server, SPP server and LE slave support. Also generating a GATT library with server support and a HID library with device role support.

Usage: fbl_ rvmdk.pl 32_M3 --sdpserver --sppserver --leslave --gattserver --hiddevice

Generates: libBluetopia_LE_32_M3.a, libSS1BTGAT_32_M3.a, libSS1BTHID_32_M3.a

Generating a core library with Processor type M4 wchar=32 with HW Floating point, SDP server and client support and LE master and slave support. Also generating a GATT library with client support and a HID library with host role support.

Usage: fbl_ rvmdk.pl 32_M4.fp_HW_FP --sdpserver --sdpclient --lemaster --leslave -- gattclient --hidhost

Generates: libBluetopia_LE_32_M4.fp_HW_FP.a, libSS1BTGAT_32_M4.fp_HW_FP.a,
libSS1BTHID_32_M4.fp_HW_FP.