RTL8762D Bluetooth LE
Sample Project User Manual

V 1.0.1

2021/11/19

Revision History

Date	Version	Comments	Author	Reviewer
2021/07/02	V1.0.0	Formal version	Herry	

Contents

Table List

Figure List

Glossary

Terms	Definitions
LE	Low Energy
GAP	Generic Access Profile

1 Overview

1.1 Upper Stack

Upper Stack implements the Host part of Bluetooth stack. Upper Stack use this method to implement: Upper Stack image.

When APP uses Upper Stack image, there is a separate Upper Stack image for download. Upper Stack and APP update independently.

Upper Stack image will be introduced in detail in the chapter *Upper Stack Image*.

1.2 Upper Stack Configuration

Upper Stack image provides Upper Stack with different configurations. The different configurations of Upper Stack files are represented by upperstack_A_B. The meanings of A and B are as follows:

A indicates the Bluetooth Technology Features that Upper Stack files supported. If Upper Stack configurations contain the same A, Upper Stack files with these configurations support the same Bluetooth Technology Features. For specific Bluetooth Technology Features supported by Upper Stack, please refer to the upperstack_config.h file. The upperstack_config.h file will be introduced in detail in the chapter *Upper Stack Features Configuration*.

B indicates the Upper Stack performance. By default, Upper Stack is configured as flash code. For specific configuration of Upper Stack files, please refer to readme file. The directory of readme file is as follows:

- Upper Stack image: sdk\bin\upperstack_img\readme.

1.3 Upper Stack Features Configuration

An upperstack_config.h file is provided in Upper Stack files for APP to use. The macro definitions corresponding to the supported Bluetooth Technology features are introduced in upperstack_config.h file. 0 means the feature is not supported, and 1 means the feature is supported. For the corresponding relationship between macro definition and Bluetooth Technology features, please refer to the chapter *Supported Bluetooth Technology Features* in the *Bluetooth LE Stack User Manual*.

When the Upper Stack supports a feature, APP can call the reference API.

For example, Upper Stack supports F_BT_LE_READ_CHANN_MAP.

```
#define F_BT_LE_READ_CHANN_MAP                    (F_BT_LE_SUPPORT && 1)
```

APP can call the reference API le_read_chann_map.

```
#if F_BT_LE_READ_CHANN_MAP
/**
  * @brief   Read the used channel map of the connection. Channel map value will be
 returned by
  *          @ref app_gap_callback with cb_type @ref GAP_MSG_LE_READ_CHANN_MAP.
  *
  * @param[in] conn_id Connection ID
  * @return  Read request result.
  * @retval  GAP_CAUSE_SUCCESS: Read request sent success.<BR>
  * @retval  GAP_CAUSE_NON_CONN: Read request sent fail.<BR>
  */
T_GAP_CAUSE le_read_chann_map(uint8_t conn_id);
#endif
```

2 Upper Stack Image

2.1 Upper Stack Image Configuration

The configurations of Upper Stack image are shown in the Table 2-1.

The upperstack_0_0 is the default configuration of Upper Stack image. Upper Stack is configured as flash code. For specific Bluetooth Technology Features supported by Upper Stack, please refer to the upperstack_config.h file.

Table 2-1 Upper Stack Image with Different Configuration

Upper Stack files	Image Directory	Flash Size	RAM Size	Reference Project
upperstack_0_0 (Default)	bin\upperstack_img\upperstack_0_0	144K	2KB	board\evb\xxxxx

The specific files contained in the Upper Stack image and the directory are described in the Table 2-2.

Table 2-2 Files List of Upper Stack Image

Upper Stack Configuration	Files of Upper Stack	Remark
	bin\upperstack_img\readme	Introduction to Upper Stack image of different configurations
upperstack_0_0 (Default)	bin\upperstack_img\upperstack_0_0\upperstack_MP_xxxx.bin	Upper Stack image
	bin\upperstack_img\upperstack_0_0\gap_utils.lib	gap lib
	bin\upperstack_img\upperstack_0_0\upperstack_config.h	Bluetooth Technology Feature configuration

The gap_utils.lib file used by the APP must be in the same directory as the Upper Stack image. The APP and Upper Stack are updated independently. If the Upper Stack image is updated, APP does not need to be recompiled.

2.2 Usage of Upper Stack Image

When Upper Stack image is used, APP should add the directory of corresponding upperstack_config.h file into the include path. Because the RAM size of Upper Stack image with different configuration may be different, APP needs to modify the mem_config.h file. For RAM size parameter, please refer to Table 2-1 in chapter *Upper Stack Image Configuration* or bin\upperstack_img\readme file.

For example, when APP uses upperstack_0_0, the RAM size of Upper Stack that obtained from readme file is 8K.

```
/*============================================================================*
 *                              upperstack_0_0                                *
 *============================================================================*/
The flash size of upperstack_0_0 image is 144K. If Application uses upperstack_0_0,
Application should configure UPPERSTACK_GLOBAL_SIZEas value which is larger than or
equal to (2 * 1024) in mem_config.h of Application.

/** @brief data ram size for upperstack global variables and code */
#define UPPERSTACK_GLOBAL_SIZE     (2 * 1024)  /* larger than or equal to (2 * 1024) */
```

The UPPERSTACK_GLOBAL_SIZE in mem_config.h file should be greater than or equal to 2K.

```
/** @brief data ram size for upperstack global variables and code */
    #define UPPERSTACK_GLOBAL_SIZE          (2 * 1024)
```

The Upper Stack image include path in the APP is shown in Figure 2-1.

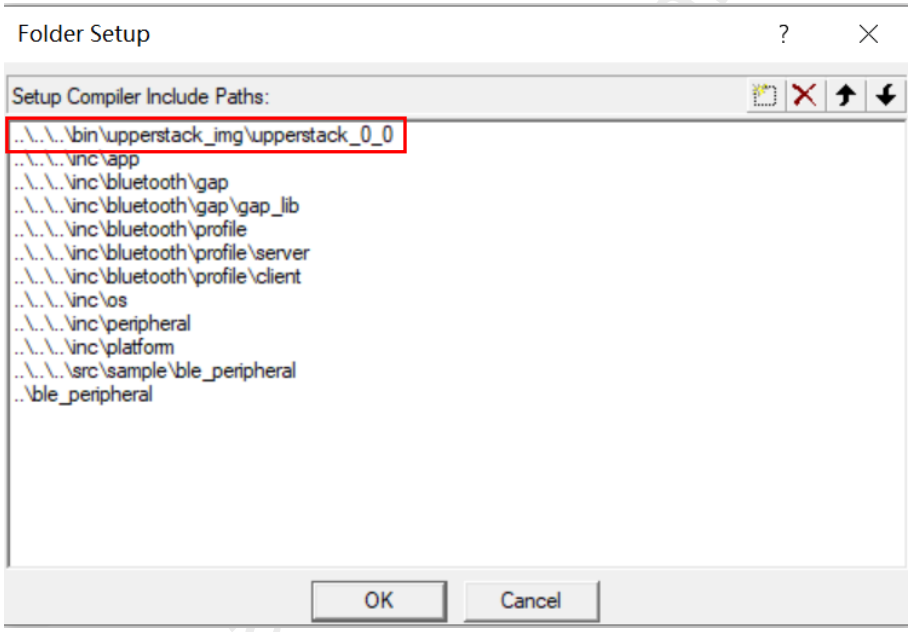

Figure 2-1 Include Path

2.3 Sample Projects

The directory of Upper Stack image sample projects is sdk\board\evb. The sample projects will be introduced in detail in the chapter *Bluetooth LE Sample* Projects.

3 GAP Lib

GAP lib provides GAP extension functions for application.

3.1 GAP Extension Function

1. **Vendor Function module**

   Vendor function module provides vendor-specific functions. More information please refers to gap_vendor.h with the directory sdk\inc\bluetooth\gap\gap_lib\gap_vendor.h.

3.2 Usage of GAP Lib

To use GAP extension function, application shall add gap_utils.lib to project.

GAP lib directory is as follows:

- Upper Stack image: sdk\bin\upperstack_img\upperstack_x_x\gap_utils.lib. The directory needs to be determined according to the configuration of the Upper Stack image used.

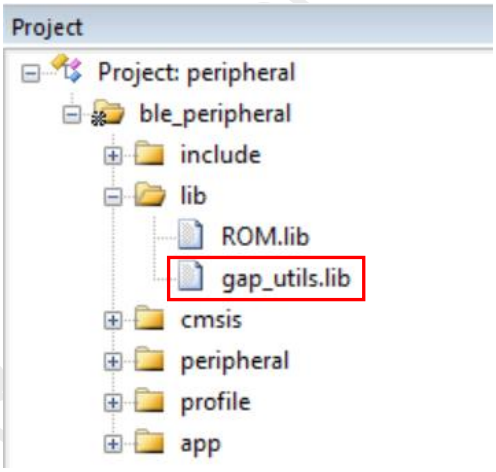

Figure 3-1 GAP Lib

And it is necessary to initialize gap_utils.lib with gap_lib_init() function before using extension function.

```
int main(void)
{
    …
    gap_lib_init();
    …
    task_init();
    os_sched_start();
```

```
    return 0;
}
```

4 Bluetooth LE Sample Projects

This chapter only introduces the project configuration of the example projects. For source code and test procedures, please refer to the chapter *Bluetooth LE example project* in the *Bluetooth LE Stack User Manual*.

Table 4-1 Bluetooth LE Sample Projects

Sample Projects	GAP Role				GATT Server	GATT Client	Link Layer in connection state		Link Number	Other functions
	Broadcaster Role	Observer Role	Peripheral Role	Central Role			Slave Role	Master Role		
Broadcaster Application	√								0	
Observer Application		√							0	
Peripheral Application			√		√	√	√		1	ANCS Client
Central Application				√	√	√		√	2 (default)	
Scatternet Application	√	√	√	√	√	√	√	√	2 (default)	Airplane mode, Set PHY
BT5 Peripheral Application			√				√		1	LE Advertising Extensions
BT5 Central Application				√				√	1	LE Advertising Extensions
Peripheral Privacy Application			√		√		√		1	LE Privacy

4.1 BLE Broadcaster Application

4.1.1 Project Overview

This section describes the project directory and project structure. Reference files directory as follows:

- Upper Stack image project directory: sdk\board\evb\ble_broadcaster

- Project source code directory: sdk\src\sample\ble_broadcaster

Directory structure of the project is shown in Figure 4-1:

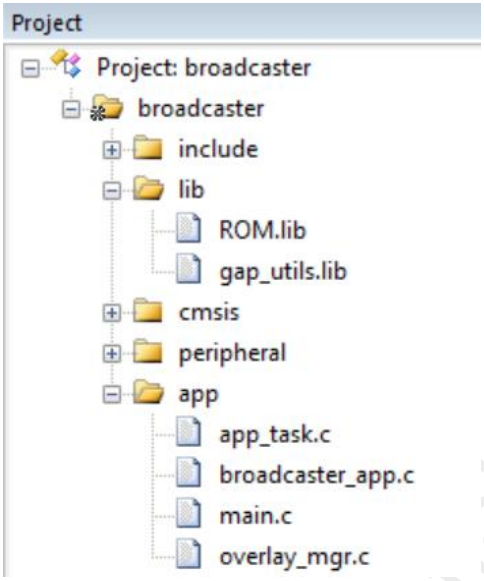

Figure 4-1 Broadcaster Project Directory Structure

The files can be divided into the following groups:

Table 4-2 Broadcaster Project File List

Directory	Description
include	rom_uuid.h: ROM UUID header files. User need not modify.
lib	The gap library file.
cmsis	The cmsis source code. User need not modify.
peripheral	The peripheral source code.
app	The application source code.

4.2 BLE Observer Application

4.2.1 Project Overview

This section describes the project directory and project structure. Reference files directory as follows:

- Upper Stack image project directory: sdk\board\evb\ble_observer
- Project source code directory: sdk\src\sample\ble_observer

Directory structure of the project is shown in Figure 4-2:

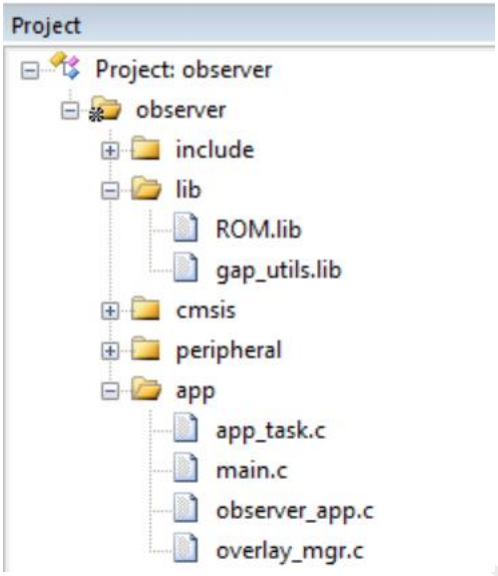

Figure 4-2 Observer Project Directory Structure

The files can be divided into the following groups:

Table 4-3 Observer Project File List

Directory	Description
include	rom_uuid.h: ROM UUID header files. User need not modify.
lib	The gap library file.
cmsis	The cmsis source code. User need not modify.
peripheral	The peripheral source code.
app	The application source code.

4.3 BLE Peripheral Application

4.3.1 Project Overview

This section describes the project directory and project structure. Reference files directory as follows:

- Upper Stack image project directory: sdk\board\evb\ble_peripheral
- Project source code directory: sdk\src\sample\ble_peripheral

Directory structure of the project is shown in Figure 4-3:

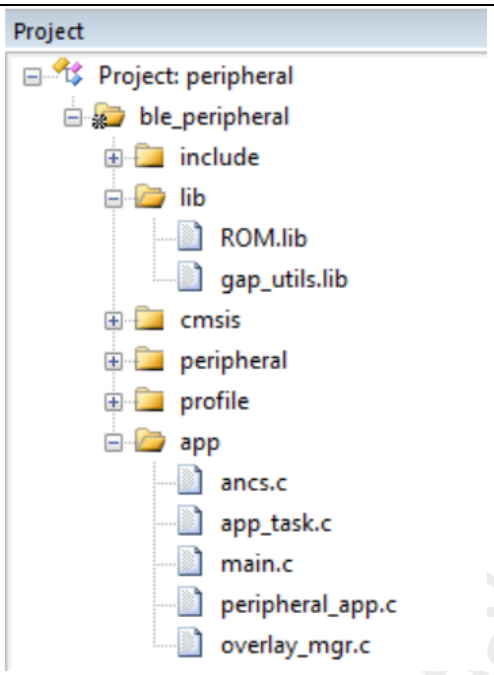

Figure 4-3 Peripheral Project Directory Structure

The file list is divided into the following groups:

Table 4-4 Peripheral Project File List

Directory	Description
include	rom_uuid.h: ROM UUID header files. User need not modify.
lib	The gap library file.
cmsis	The cmsis source code. User need not modify.
peripheral	The peripheral source code.
profile	The GATT profiles source code.
app	The application source code.

4.4 BLE Central Application

4.4.1 Project Overview

This section describes the project directory and project structure. Reference files directory as follows:

- Upper Stack image project directory: sdk\board\evb\ble_central
- Project source code directory: sdk\src\sample\ble_central

Directory structure of the project is shown in Figure 4-4:

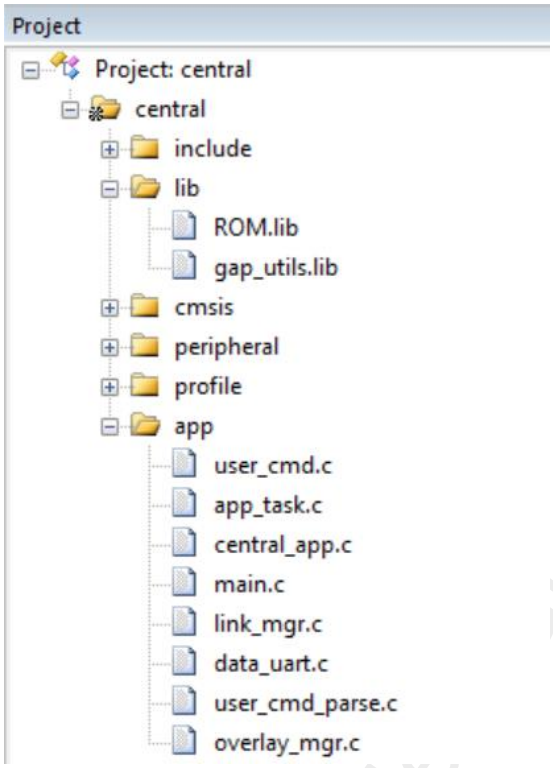

Figure 4-4 Central Project Directory Structure

The file list is divided into the following groups:

Table 4-5 Central Project File List

Directory	Description
include	rom_uuid.h: ROM UUID header files. User need not modify.
lib	The gap library file.
cmsis	The cmsis source code. User need not modify.
peripheral	The peripheral source code.
profile	The GATT profiles source code.
app	The application source code.

4.5 BLE Scatternet Application

4.5.1 Project Overview

This section describes the project directory and project structure. Reference files directory as follows:

- Upper Stack image project directory: sdk\board\evb\ble_scatternet
- Project source code directory: sdk\src\sample\ble_scatternet

Directory structure of the project is shown in Figure 4-5:

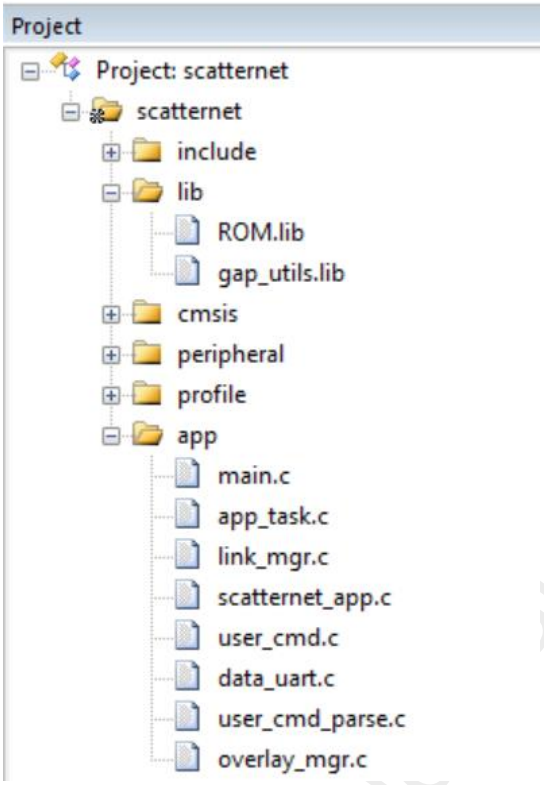

Figure 4-5 Scatternet Project Directory Structure

The file list is divided into the following groups:

Table 4-6 Scatternet Project File List

Directory	Description
include	rom_uuid.h: ROM UUID header files. User need not modify.
lib	The gap library file.
cmsis	The cmsis source code. User need not modify.
peripheral	The peripheral source code.
profile	The GATT profiles source code.
app	The application source code.

4.6 BLE BT5 Peripheral Application

4.6.1 Project Overview

This section describes the project directory and project structure. Reference files directory as follows:

- Upper Stack image project directory: sdk\board\evb\ble_bt5_peripheral
- Project source code directory: sdk\src\sample\ble_bt5_peripheral

Directory structure of the project is shown in Figure 4-6:

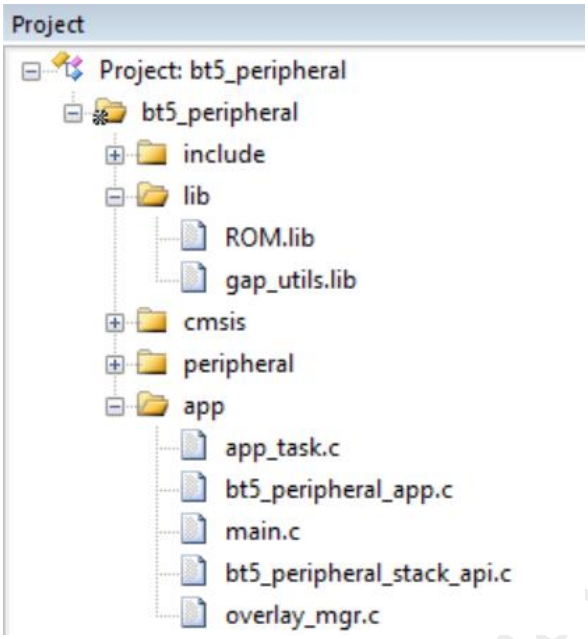

Figure 4-6 BT5 Peripheral Project Directory Structure

The file list is divided into the following groups:

Table 4-7 BT5 Peripheral Project File List

Directory	Description
include	rom_uuid.h: ROM UUID header files. User need not modify.
lib	The gap library file.
cmsis	The cmsis source code. User need not modify.
peripheral	The peripheral source code.
app	The application source code.

4.7 BLE BT5 Central Application

4.7.1 Project Overview

This section describes the project directory and project structure. Reference files directory as follows:

- Upper Stack image project directory: sdk\board\evb\ble_bt5_central
- Project source code directory: sdk\src\sample\ble_bt5_central

Directory structure of the project is shown in Figure 4-7:

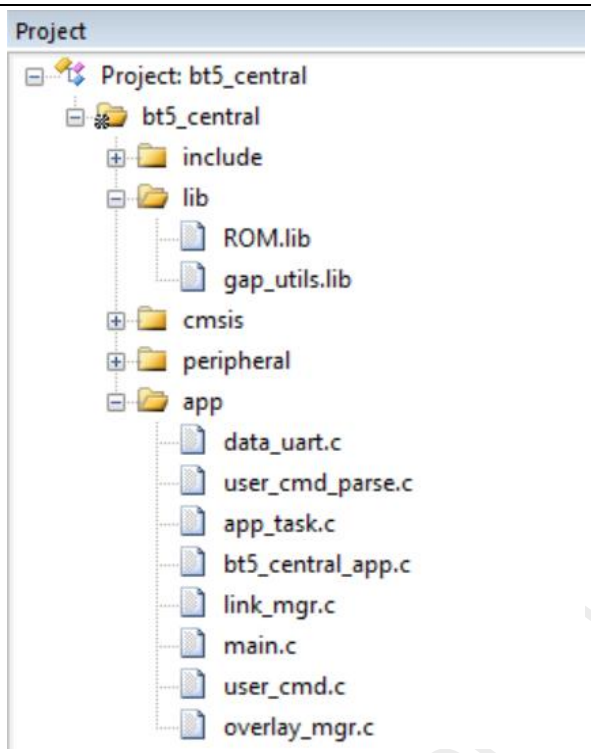

Figure 4-7 BT5 Central Project Directory Structure

The file list is divided into the following groups:

Table 4-8 BT5 Central Project File List

Directory	Description
include	rom_uuid.h: ROM UUID header files. User need not modify.
lib	The gap library file.
cmsis	The cmsis source code. User need not modify.
peripheral	The peripheral source code.
app	The application source code.

4.8 BLE Peripheral Privacy Application

4.8.1 Project Overview

This section describes the project directory and project structure. Reference files directory as follows:

- Upper Stack image project directory: sdk\board\evb\ble_peripheral_privacy
- Project source code directory: sdk\src\sample\ble_peripheral_privacy

Directory structure of the project is shown in Figure 4-8:

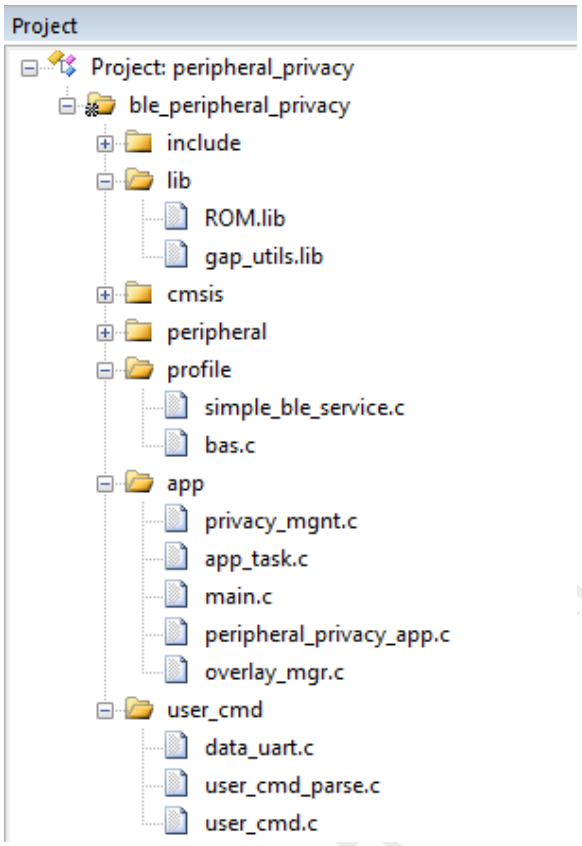

Figure 4-8 Peripheral Privacy Project Directory Structure

The file list is divided into the following groups:

Table 4-9 Peripheral Privacy Project File List

Directory	Description
include	rom_uuid.h: ROM UUID header files. User need not modify.
lib	The Upper Stack files and gap library file.
cmsis	The cmsis source code. User need not modify.
peripheral	The peripheral source code.
profile	The GATT profiles source code.
app	The application source code.
user_cmd	The user command source code.

References

[1] Bluetooth SIG. Core_v5.2 [M]. 2019.

[2] Bluetooth LE Stack User Manual