USB Stack OTG Reference Manual

Document Number: USBSOTGRM

Rev. 2, 09/2015

Contents

Chapter 1 Before You Begin

1.1 About this book

This *USB Stack OTG Reference Manual* describes the USB OTG driver and the programming interface in the USB Stack.

The audience should be familiar with the following reference material:

- *Universal Serial Bus Specification Revision 1.1*
- *Universal Serial Bus Specification Revision 2.0*
- *On-The-Go and Embedded Host Supplement to the USB Revision 2.0 Specification Revision 2.0 version 1.1a*

Use this book in addition to:

- *Source Code*

1.2 Acronyms and abbreviations

Table 1 Acronyms and abbreviations

Term	Description
API	Application Programming Interface
CDC	Communication Device Class
HID	Human Interface Device
MSD	Mass Storage Device
PHDC	Personal Healthcare Device Class
OTG	On The Go
SRP	Session Request Protocol
HNP	Host Negotiation Protocol

1.3 Function listing format

This is the general format of an entry for a function, compiler intrinsic, or a macro.

function_name()

A short description of what function **function_name()** does.

Synopsis

Provides a prototype for function **function_name()**.

```
<return_type> function_name(
<type_1> parameter_1,
<type_2> parameter_2,
...
<type_n> parameter_n)
```

Parameters

parameter_1 [in] – Pointer to x

parameter_2 [out] – Handle for y

parameter_n [in/out] – Pointer to z

Parameter passing is categorized as follows:

- *In* – indicates that the function uses one or more values in the parameter you give it without storing any changes.
- *Out* – indicates that the function saves one or more values in the parameter you give it. Examine the saved values to find out useful information about your application.
- *In/out* – indicates the function changes one or more values in the parameter you give it and saves the result. Examine the saved values to find out useful information about your application.

Description – Describes the function **function_name()**. This section also describes any special characteristics or restrictions that might apply:

- Function blocks or might block under certain conditions
- Function must be started as a task
- Function creates a task
- Function has pre-conditions that might not be obvious
- Function has restrictions or special behavior

Return value – Specifies any value or values returned by function **function_name()**.

See also – Lists other functions or data types related to function **function_name()**.

Example – Provides an example (or a reference to an example) that illustrates the use of function **function_name()**.

Chapter 2 Overview

2.1 USB overview

Universal Serial Bus (USB) is a serial bus protocol initially designed to connect external devices to a host computer. The main role of the host is to coordinate the connected devices and to manage all communications in a USB bus. Due to the increase of the number of USB components with different capabilities and functionalities, the current trend is to extend the possibility to connect and use devices directly, without PC intervention. Several typical applications are: a photo camera can be connected directly to a printer, an audio devices (such as an MP3 player) can search and play music titles directly from a mass storage device attached to it, a digital photo camera can store pictures or movies directly onto a mass storage device, and so on. This role is accomplished by an extension to the USB stack called On-The-Go (OTG).

OTG accomplishes two major functions:

· Swap the host role between the two devices connected, using an OTG proprietary protocol called HNP.

· In an OTG connection, one device (called device A) is responsible to provide voltage on the VBUS line; typically the bus voltage provided is not continuous (thus allowing it to conserve power); the device supplied on the bus (known as device B) can require that the provider device A powers up the USB bus using a special protocol named SRP.

The USB OTG software consists of the following:

- USB OTG application
- USB OTG driver
- USB OTG controller interface - low-level functions used to interact with the USB OTG controller hardware
- USB Host software
- USB Device software
- OS adapter to provide unified OS API to USB Stack
- SoC-specific initialization.

  In <install_dir>/usb/usb_core/otg/sources/bsp/SOC_NAME/usb_otg_bsp.c

  ```
  usb_status usb_otg_soc_init(uint8_t controller_id)
  ```

  ```
  The controller_id is the enum CONTROLLER_INDEX
  ```

- Board-specific initialization.

  In <install_dir>/examples/BOARD_NAME /board.c

  ```
  uint8_t usb_otg_board_init(uint8_t controller_id)
  ```

```
The controller_id is the enum CONTROLLER_INDEX
```

The architecture and components of the USB stack are as follows:

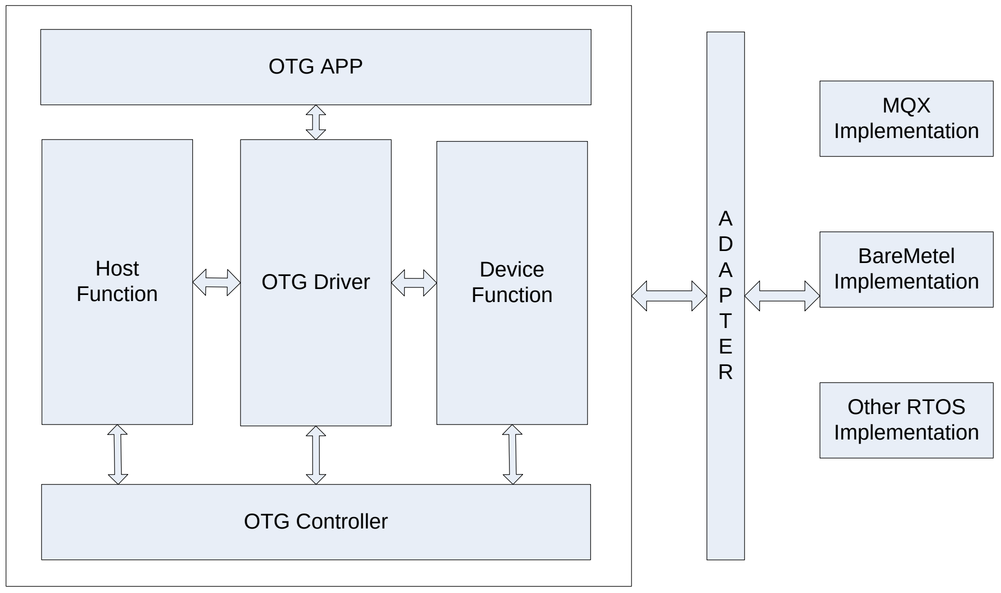

Figure 1 USB OTG stack architecture

2.2 Using the USB OTG API

To use the USB OTG API, follow these general steps. Each API function is described in the subsequent chapters.

1. Initialize the USB OTG controller interface (usb_otg_init()).
2. Shut down the USB OTG controller interface (usb_otg_shut_down()).
3. Get device state, role A or B (usb_otg_get_state()).
4. Role B sends a session request signal (usb_otg_session_request()).
5. Role B sends a HNP signal (usb_otg_bus_request()).
6. Role B release USB Bus (usb_otg_bus_release()).
7. Role A set USB Bus request status (usb_otg_set_a_bus_req()).
8. Role A get USB Bus request status (usb_otg_get_a_bus_req()).
9. Role A set USB Bus drop status (usb_otg_set_a_bus_drop()).
10. Role A get USB Bus drop status (usb_otg_get_a_bus_drop()).
11. Role A clear error status (usb_otg_set_a_clear_err()).

2.3 API overview

This section describes the USB Stack OTG API functions.

Table 2 USB OTG controller driver APIs

No.	API function	Description
1	usb_otg_init()	Initialize the USB OTG controller
2	usb_otg_shut_down()	Shut down the USB OTG controller
3	usb_otg_get_state()	Get device state , role A or B
4	usb_otg_session_request()	Role B sends a session request(SRP) signal
5	usb_otg_bus_request()	Role B sends a HNP signal
6	usb_otg_bus_release()	Role B release USB Bus
7	usb_otg_set_a_bus_req()	Role A set USB Bus request status
8	usb_otg_get_a_bus_req()	Role A get USB Bus request status
9	usb_otg_set_a_bus_drop()	Role A set USB Bus drop status
10	usb_otg_get_a_bus_drop()	Role A get USB Bus drop status
11	usb_otg_set_a_clear_err()	Role A clear error status

Chapter 3 USB OTG APIs

3.1 usb_otg_init

Synopsis

```
usb_status usb_otg_init
(
    uint8_t controller_id,
    otg_int_struct_t * init_struct_ptr,
    usb_otg_handle *  handle
)
```

Parameters

controller_id	[in]	– controller ID
		KHCI 0 --- 0
		KHCI 1 --- 1
init_struct_ptr	[in]	– OTG initialization structure
handle	[out]	– OTG handle

Description

This function should be called prior to any other function of the OTG API. It verifies the input parameters and if they are correct it allocates memory for the usb_otg_state_struct_t, initializes the structure, passes the pointer to this structure to application through the handle parameter, and initializes the internal (on chip) and external OTG hardware. For user's board, the function bsp_usb_otg_set_peripheral_init_param() must be called before this function called, see *Freescale KSDK USB Stack Porting New Platform User's Guide* (document USBSPNPUG) for more information.

Return Value

USB_OK (success)

Others (failure)

3.2 usb_otg_shut_down

Synopsis

```
usb_status usb_otg_shut_down
(
    usb_otg_handle otg_handle
)
```

Parameters

otg_handle	[in] – OTG handle

Description

The function is used to stop the OTG controller.

Return Value

USB_OK (success)

Others (failure)

3.3 usb_otg_get_state

Synopsis

```
uint8_t usb_otg_get_state
(
usb_otg_handle otg_handle
)
```

Parameters

otg_handle	[in] – OTG handle

Description

The function is used to get the device role state.

Return Value

0x00 USB_OTG_DEVSTATE_B

0x01 USB_OTG_DEVSTATE_A

0xFF USB_OTG_DEVSTATE_UNDEFINED

3.4 usb_otg_session_request

Synopsis

```
usb_status usb_otg_session_request
(
usb_otg_handle handle
)
```

Parameters

handle	[in] – OTG handle

Description

The USB OTG session request function is used to start the SRP protocol from the B-Device side with the target of requesting the Vbus from the A-Device.

The function is used to send a SRP signal and can only be called by the B role device.

Return Value

USB_OK (success)

Others (failure)

3.5 usb_otg_bus_request

Synopsis

```
usb_status usb_otg_bus_request
(
usb_otg_handle handle
)
```

Parameters

handle	[in] – OTG handle

Description

The USB OTG bus request function is used to start the HNP protocol from the B-Device side with the target of obtaining the bus control from the A-Device (B-Peripheral will become a B-Host and A-Host becomes A-Peripheral). This request can only be initiated from the B-device state.

Return Value

USB_OK (success)

Others (failure)

3.6 usb_otg_bus_release

Synopsis

```
usb_status usb_otg_bus_release
(
usb_otg_handle handle
)
```

Parameters

handle	[in] – OTG handle

Description

The USB OTG bus request function is used to end a Host session on the B-device. This request is only accepted if the device is in the B-host state and results in the B device releasing the bus and returning in the B-peripheral state.

Return Value

USB_OK (success)

Others (failure)

3.7 usb_otg_set_a_bus_req

Synopsis

```
usb_status usb_otg_set_a_bus_req
(
usb_otg_handle otg_handle
bool            a_bus_req
)
```

Parameters

otg_handle	[in] – OTG handle
a_bus_req	[in] – Request USB Bus status

Description

The function is used to set USB Bus request status and can only be called by the A role device.

Return Value

USB_OK (success)

Others (failure)

3.8 usb_otg_get_a_bus_req

Synopsis

```
usb_status usb_otg_get_a_bus_req
(
usb_otg_handle otg_handle
bool           *a_bus_req
)
```

Parameters

otg_handle	[in] – OTG handle
a_bus_req	[out] – Request USB Bus status

Description

The function is used to get USB Bus request status and can only be called by the A role device.

Return Value

USB_OK (success)

Others (failure)

3.9 usb_otg_set_a_bus_drop

Synopsis

```
usb_status usb_otg_set_a_bus_req
(
usb_otg_handle otg_handle
bool            a_bus_drop
)
```

Parameters

otg_handle	[in] – OTG handle
a_bus_drop	[in] – set USB Bus drop status

Description

The function is used to set USB Bus drop status and can only be called by the A role device.

Return Value

USB_OK (success)

Others (failure)

3.10 usb_otg_get_a_bus_drop

Synopsis

```
usb_status usb_otg_get_a_bus_drop
(
usb_otg_handle otg_handle
bool          *a_bus_drop
)
```

Parameters

otg_handle	[in] – OTG handle
a_bus_drop	[out] – USB Bus drop status

Description

The function is used to get USB Bus drop status and can only be called by the A role device.

Return Value

USB_OK (success)

Others (failure)

3.11 usb_otg_set_a_clear_err

Synopsis

```
usb_status usb_otg_set_a_clear_err
(
usb_otg_handle otg_handle
)
```

Parameters

otg_handle	[in] – OTG handle

Description

The function is used to clear an error and can only be called by the A role device.

Return Value

USB_OK (success)

Others (failure)

Chapter 4 USB OTG Configuration

4.1 USBCFG_OTG_USE_IRC48M

1 indicates IRC 48M clock source is enabled.

0 indicates IRC 48M clock source is disabled.

4.2 USBCFG_OTG_TASK_PRIORITY

The priority of OTG task.

Note:

1. For RTOS, application unblocks tasks' priorities. They should be lower than the above priority.

2. When the OTG is a host, the host tasks are created. For more information, see the *USB Stack Host Reference Manual* (document USBSHRM).

3. When the OTG works as a device, the device tasks are created. See the *USB Stack Device Reference Manual* (document USBSDRM).

Chapter 5 Revision history

This table summarizes revisions to this document since the release of the previous version

Revision History		
Revision number	**Date**	**Substantive changes**
1	04/2015	KSDK 1.2.0 Release
2	09/2015	Updated Section 2.1

Document Number: USBSHRM
Rev. 2
09/2014