ANALOG DEVICES

USB CORE API DOCUMENTATION

DATE: 07/12/2007

Table of Contents

adi_usb_CoreInit

Description
This API is used to initialize the internal structures of the USB core. This API has to be called before using any other core API functions. Typically class drivers issue this call to initialize the core. Multiple class drivers may call this API, but the initialization is done only once.

Prototype

```
ADI_USB_RESULT adi_usb_CoreInit(

    void        *pConfigData

);
```

Arguments

pConfigData	Pointer to the configuration data. This parameter is currently not used.

Return Value

In debug mode, this routine returns one of the following result codes; otherwise, ADI_USB_RESULT_SUCCESS is always returned.

ADI_USB_RESULT_SUCCESS	Successfully initialized the USB core.
ADI_USB_RESULT_FAILED	USB Core is failed to initialize

Operation Mode

This API has to be called to initialize the core in both modes before issuing any other core API calls.

MODE	**Support**
Device Mode	Yes
OTG Host Mode	Yes

adi_usb_CreateDevice

Description
This API is used to create a USB peripheral device object. Only one device object can exist for a USB device. USB peripheral devices use this API to create device object. More than one class driver may open the underlying USB peripheral driver but only one instance of the device object exists.

Prototype

```
s32_t adi_usb_CreateDevice(

    PDEVICE_OBJECT          *pDeviceObject

);
```

Arguments

pDeviceObject	Double pointer to the DEVICE_OBJECT

Return Value

Upon success device object ID is returned and the pointer to the device object is set in the input argument pDeviceObject.

Upon failure this API returns ADI_USB_RESULT_INVALID_ID.

Operation Mode

This API is used by the USB peripheral drivers in both device mode and OTG host mode to create the USB device object.

MODE	**Support**
Device Mode	Yes
OTG Host Mode	Yes

adi_usb_CreateConfiguration

Description
This API is used to create a class configuration object. In device mode class driver use this API to create one or more configurations. In case of OTG host mode USB core uses this API to represent the configuration(s) supported by the device.

Prototype

```
s32_t adi_usb_CreateConfiguration(

    PCONFIG_OBJECT          *pConfigObject

);
```

Arguments

pConfigObject	Double pointer to the PCONFIG_OBJECT

Return Value

Upon success configuration object ID is returned and the pointer to the configuration object is set in the input argument pConfigObject.

Upon failure this API returns ADI_USB_RESULT_INVALID_ID.

Operation Mode

MODE	**Support**
Device Mode	Yes
OTG Host Mode	Yes

adi_usb_CreateInterface

Description
This API is used to create a USB Interface object. In device mode class drivers create interfaces for a specific configuration. In case of OTG Host USB core uses this API to represent the interfaces supported by the device.

Prototype

```
s32_t adi_usb_CreateInterface(

    PINTERFACE_OBJECT          *pInterfaceObject

);
```

Arguments

pInterfaceObject	Double pointer to the INTERFACE_OBJECT

Return Value

Upon success interface object ID is returned and the pointer to the interface object is set in the input argument pInterfaceObject.

Upon failure this API returns ADI_USB_RESULT_INVALID_ID.

Operation Mode

MODE	**Support**
Device Mode	Yes
OTG Host Mode	Yes

adi_usb_CreateEndpoint

Description
This API is used to create an endpoint object. In device mode class drivers creates one or more endpoints for the supported interfaces. In case of OTG Host mode USB Core uses this API to represent the endpoints supported by the device.

Prototype

```
s32_t adi_usb_CreateEndpoint(

    PENDPOINT_OBJECT *pEndpoingObject,

    LOGICAL_EP_INFO *pLogicalEpInfo

);
```

Arguments

pEndpointObject	Double pointer to the PENDPOINT_OBJECT
pLogicalEpInfo	Logical Endpoint info

Return Value

Upon success endpoint object ID is returned and the pointer to the endpoing object is set in the input argument pEndpointObject.

Upon failure this API returns ADI_USB_RESULT_INVALID_ID.

Operation Mode

MODE	**Support**
Device Mode	Yes
OTG Host Mode	Yes

adi_usb_CreateString

Description
This API is used to create a USB string object. Class drivers use this call to create string objects by passing the ASCII strings.

Prototype

s32_t adi_usb_CreateString(

    const char   *pszAsciiString

);

Arguments

pszAsciiString	Pointer to ASCII string

Return Value

Upon success string object ID is returned.

Upon failure this API may return ADI_USB_RESULT_INVALID_ID or ADI_USB_RESULT_INVALID_INPUT.

Operation Mode

MODE	**Support**
Device Mode	Yes
OTG Host Mode	No

adi_usb_AttachConfiguration

Description
This API is used to bind a configuration object with the device object. In case of device mode class drivers use this API to bind a configuration with the device. In case of OTG Host mode this API is used to bind the obtained configuration information with the device object.

Prototype

```
ADI_USB_RESULT adi_usb_AttachConfiguration(

        const s32_t dwDeviceID,

         const s32_t dwConfigID

);
```

Arguments

dwDeviceID	Device ID that's is returned via adi_usb_CreateDevice call
dwConfigID	Configuration ID that's is returned through adi_usb_CreateConfiguraiton call

Return Value

In debug mode, this routine returns one of the following result codes; otherwise, ADI_USB_RESULT_SUCCESS is always returned.

ADI_USB_RESULT_SUCCESS	API call is successful
ADI_USB_RESULT_FAILED	Failed to bind the configuration

Operation Mode

MODE	**Support**
Device Mode	Yes
OTG Host Mode	Yes

adi_usb_AttachInterface

Description
This API is used to bind an Interface Object with a Configuration Object. In case of device mode class drivers use this API to bind an interface with an configuration. In case of OTG Host mode this API is used to bind the obtained interface information with the associated configuration.

Prototype

```
ADI_USB_RESULT adi_usb_AttachInterface(

    const s32_t dwConfigID,

     const s32_t dwInterfaceID

);
```

Arguments

dwConfigID	Configuration ID that's is returned via adi_usb_CreateConfiguration call
dwInterfaceID	Interface ID that's is returned through adi_usb_CreateInterface call

Return Value

In debug mode, this routine returns one of the following result codes; otherwise, ADI_USB_RESULT_SUCCESS is always returned.

ADI_USB_RESULT_SUCCESS	API call is successful
ADI_USB_RESULT_FAILED	Failed to bind the interface

Operation Mode

MODE	**Support**
Device Mode	Yes
OTG Host Mode	Yes

adi_usb_AttachEndpoint

Description
This API is used to bind an endpoint object with an interface object. In case of device mode class drivers use this API to bind an endpoint with an interface. In case of OTG Host mode this API is used to bind the obtained endpoint information with the associated interface.

Prototype

```
ADI_USB_RESULT adi_usb_AttachEndpoint(

    const s32_t dwInterfaceID,

     const s32_t dwEndpointID

);
```

Arguments

dwInterfaceID	Interface ID that's is returned via adi_usb_CreateInterface call
dwEndpointID	Endpoint ID that's is returned through adi_usb_CreateEndpoint call

Return Value

In debug mode, this routine returns one of the following result codes; otherwise, ADI_USB_RESULT_SUCCESS is always returned.

ADI_USB_RESULT_SUCCESS	API call is successful
ADI_USB_RESULT_FAILED	Failed to bind the configuration

Operation Mode

MODE	**Support**
Device Mode	Yes
OTG Host Mode	Yes

adi_usb_GetDeviceDescriptor

Description
This API returns pointer to the device descriptor object. This API is used by the class drivers to get the device descriptor object and to set fields like Vendor ID, Product ID, and Serial Number.

Prototype

PDEVICE_DESCRIPTOR adi_usb_GetDeviceDescriptor(void);

Arguments
void

Return Value

Returns the device descriptor object.

Operation Mode

MODE	**Support**
Device Mode	Yes
OTG Host Mode	No

adi_usb_SetPhysicalEndpointInfo

Description
This API is used by the peripheral drivers to supply the capabilities of the physical endpoints to the core. This information is used by the core to allocate best suitable endpoint for the class driver's adi_usb_CreateEndpoint requests. Typical information includes the endpoint FIFO size.

Prototype

```
void adi_usb_SetPhysicalEndpointInfo(

    const s32_t wNumEndPoints,

    PHYSICAL_EP_INFO PhysicalEndpointInfo[]

);
```

Arguments

wNumEndPoints	Number of Physical endpoints
PhysicalEndpointInfo	Each entry in this array corresponds to the endpoint information. Starting with endpoint 0.

Return Value
Void function

Operation Mode

MODE	**Support**
Device Mode	Yes
OTG Host Mode	Yes

adi_usb_SetPhysicalDriverInfo

Description
This API is used by the class drivers to pass the physical driver information to the core. Core works independent of the underlying physical driver. So class drivers have to provide physical driver information by supplying the entry point and the physical driver handle.

Prototype

void adi_usb_SetPhysicalDriverInfo(

ADI_DEV_PDD_ENTRY_POINT *pEntryPoint,

ADI_DEV_PDD_HANDLE *pPDDHandle);

Arguments

pEntryPoint	Physical Driver Entrypoint
pPDDHandle	Pointer to the already opened physical driver handle.

Return Value
Void function

Operation Mode

MODE	**Support**
Device Mode	Yes
OTG Host Mode	Yes

adi_usb_GetObjectFromID

Description
This API is used get the pointer to the object that is identified by the given ID.

Prototype

void adi_usb_GetObjectFromID(

const s32_t dwObjectID,

USB_OBJECT_TYPE eObjectType,

void **pObject

);

Arguments

dwObjectID	ID of the object
eObjectType	Valid Object Types: USB_DEVICE_OBJECT_TYPE USB_CONFIGURATION_OBJECT_TYPE USB_INTERFACE_OBJECT_TYPE USB_ENDPOINT_OBJECT_TYPE
pObject	Double pointer to the object

Return Value
Upon Success this API returns ADI_USB_RESULT_SUCCESS and sets the pointer pObject to point to the requested object.

Upon failure this API returns ADI_USB_RESULT_FAILED

Operation Mode

MODE	**Support**
Device Mode	Yes
OTG Host Mode	Yes

adi_usb_GetDeviceMode

Description
This API is used by the class drivers to get the current operational mode of the device.

Prototype

DEV_MODE adi_usb_GetDeviceMode(void

);

Arguments
void

Return Value
MODE_DEVICE is returned in case USB is operating in peripheral mode.
MODE_OTG_HOST is returned in case USB is operating in OTG host mode.

Operation Mode

MODE	**Support**
Device Mode	Yes
OTG Host Mode	Yes

adi_usb_SetDeviceMode

Description
This API is used by the class drivers to set the operational mode of the USB device. Typically this API is used by the OTG host application to set the mode of operation to host mode from the default device mode.

Prototype

```
s32_t adi_usb_SetDeviceMode(

    DEV_MODE    eDevMode

);
```

Arguments

eDevMode	Valid Device Mode types: MODE_DEVICE MODE_OTG_HOST

Return Value
Upon success ADI_USB_RESULT_SUCCESS is returned. And upon failure ADI_USB_RESULT_FAILED is returned.

Operation Mode

MODE	**Support**
Device Mode	Yes
OTG Host Mode	Yes

adi_usb_otg_SetConfiguration

Description
This API is used to select a configuration from the list of configurations of the device. Typically used by the OTG class drivers to select a configuration upon completion of the device enumeration.ADI_USB_OTG_EVENT_ENUMERATION_COMPLETE callback event returns the device object and all supported configurations.

Prototype

```
s32_t adi_usb_otg_SetConfiguration(

    const s32_t          dwConfigID

);
```

Arguments

dwConfigID	Selected configuration value

Return Value

Returns ADI_USB_RESULT_SUCESS upon success.

Returns ADI_USB_RESULT_FAILED upon failure.

Operation Mode

MODE	**Support**
Device Mode	No
OTG Host Mode	Yes

adi_usb_otg_SetInterface

Description
This API is used to select an alternate setting of an interface. The associated adi_usb_otg_SetConfiguration must be completed successfully before this API. The interface and alternate settings must be selected from the supported list that is given via the ADI_USB_OTG_EVENT_ENUMERATION_COMPLETE callback event.

Prototype

```
s32_t adi_usb_otg_SetInterface(
    const s32_t         dwInterfaceID,
    const s32_t         dwAltSetting
);
```

Arguments

dwInterfaceID	Interface ID
dwAltSetting	Alternate Setting of the interface ID specified by dwInterfaceID

Return Value

Returns ADI_USB_RESULT_SUCESS upon success.

Returns ADI_USB_RESULT_FAILED upon failure.

Operation Mode

MODE	**Support**
Device Mode	No
OTG Host Mode	Yes

adi_usb_otg_SetEpZeroCallback

Description
This API is used to set control endpoint callback in case of OTG host mode. This callback routine gets called once host completes the enumeration.

Prototype

```
s32_t adi_usb_otg_SetEpZeroCallback(

    const s32_t dwConfigID,

   ADI_DCB_CALLBACK_FN CallBack

);
```

Arguments

dwConfigID	Selected configuration value (default value is 1)
ADI_DCB_CALLBACK_FN	Callback function

Return Value

Returns ADI_USB_RESULT_SUCESS upon success.

Returns ADI_USB_RESULT_FAILED upon failure.

Operation Mode

MODE	**Support**
Device Mode	No
OTG Host Mode	Yes

adi_usb_ShowEventLog

Description
This API is used to show the list of logged USB events.

Prototype

void adi_usb_ShowEventLog(void

  );

Arguments

void
Return Value

void

Operation Mode

MODE	**Support**
Device Mode	Yes
OTG Host Mode	Yes