



USB Device Low Level Driver for Atmel SAM x70 User Guide

Version 1.10

For use with USB Device Low Level Driver for Atmel[®] SAM x70 versions 1.01 and above

(This supports Atmel[®] ATSAME70, ATSAMS70, ATSAMV70, and ATSAMV71 micro-controllers.)

Exported on 10/19/2018

All rights reserved. This document and the associated software are the sole property of HCC Embedded. Reproduction or duplication by any means of any portion of this document without the prior written consent of HCC Embedded is expressly forbidden.

HCC Embedded reserves the right to make changes to this document and to the related software at any time and without notice. The information in this document has been carefully checked for its accuracy; however, HCC Embedded makes no warranty relating to the correctness of this document.

Table of Contents

1	System Overview.....	3
1.1	Introduction	4
1.2	Feature Check	5
1.3	Packages and Documents	6
	Packages.....	6
	Documents	6
1.4	Change History	7
2	Source File List	8
2.1	Configuration Files.....	8
2.2	Source Code Files.....	8
2.3	Version File	8
2.4	Platform Support Package (PSP) Files.....	9
3	Configuration Options	10
3.1	config_usbd_atmel_usbhs.h.....	10
3.2	config_usbd_atmel_usbhs.c	11
4	Integration.....	12
4.1	OS Abstraction Layer	12
4.2	PSP Porting	13
	usbd_hw_init.....	14
	usbd_hw_delete	15

1 System Overview

This chapter contains the fundamental information for this module.

The component sections are as follows:

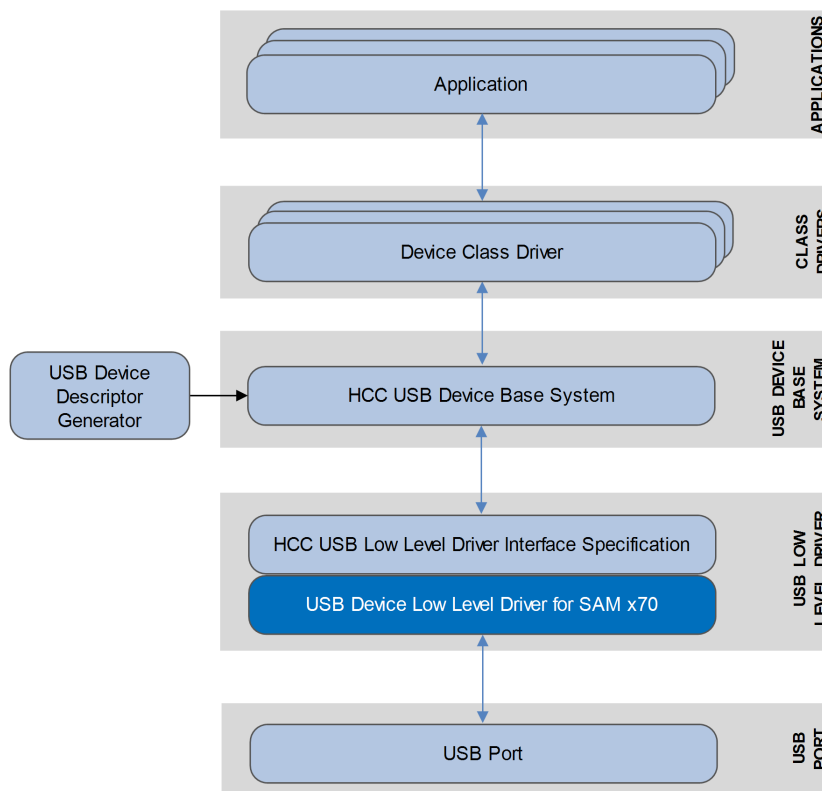
- [Introduction](#) – describes the main elements of the module.
- [Feature Check](#) – summarizes the main features of the module as bullet points.
- [Packages and Documents](#) – the *Packages* section lists the packages that you need in order to use this module. The *Documents* section lists the relevant user guides.
- [Change History](#) – lists the earlier versions of this manual, giving the software version that each manual describes.

1.1 Introduction

This guide is for those who want to configure and use the HCC Embedded Low Level Driver for Atmel[®] SAM x70 module with HCC's USB device stack. This module provides a USB device driver for Atmel[®] ATSAME70, ATSAMS70, ATSAMV70, and ATSAMV71 micro-controllers. These are used for USB High Speed peripherals.

The driver can handle all USB transfer types and, in conjunction with the USB device stack, can be used with any USB device class driver.

This package provides a low level driver for a USB stack, as shown below.



The low level driver is always started automatically by the USB device stack. The driver is linked to the stack at compile time because each low level driver uses the same function names. This also means that only one driver can run in a system.

1.2 Feature Check

The main features of the low level driver are the following:

- Conforms to the HCC Advanced Embedded Framework.
- Designed for integration with both RTOS and non-RTOS based systems.
- Conforms to HCC's USB Device Low Level Driver Specification.
- Integrated with the HCC USB device stack and all its class drivers.
- Can operate as USB High Speed or USB Full Speed driver.
- Supports all USB transfer types: control, bulk, interrupt, and isochronous.
- Provides a USB device driver for Atmel® ATSAME70, ATSAMS70, ATSAMV70, and ATSAMV71 micro-controllers. These are used for USB High Speed peripherals.

1.3 Packages and Documents

Packages

This table lists the packages that you need in order to use this module:

Package	Description
hcc_base_doc	This contains the two guides that will help you get started.
usbd_base	The USB device base package. Its source code includes the USB Driver device core.
usbd_drv_atmel_usbhs	The Atmel® SAM x70 low level driver package described by this document.
util_hcc_mem	The HCC memory management utility.

Documents

For an overview of HCC's embedded USB stacks, see [Product Information](#) on the main HCC website.

Readers should note the points in the [HCC Documentation Guidelines](#) on the HCC documentation website.

HCC Firmware Quick Start Guide

This document describes how to install packages provided by HCC in the target development environment. Also follow the *Quick Start Guide* when HCC provides package updates.

HCC Source Tree Guide

This document describes the HCC source tree. It gives an overview of the system to make clear the logic behind its organization.

HCC Embedded USB Device Base System User Guide

This document defines the USB device base system upon which the complete USB stack is built.

USB Device Low Level Driver for Atmel® SAM x70 User Guide

This is this document.

1.4 Change History

To view or download manuals, see [USB Device PDFs](#).

For the history of changes made to the package code itself, see [History: usbd_drv_atmel_usbhs](#).

The current version of this manual is 1.10. The full list of versions is as follows:

Manual version	Date	Software version	Reason for change
1.10	2018-10-19	1.01	Improved <i>PSP Porting</i> section: added two functions.
1.00	2018-05-24	1.01	First online release.

2 Source File List

This section describes all the source code files included in the system. These files follow the HCC Embedded standard source tree system, described in the *HCC Source Tree Guide*. All references to file pathnames refer to locations within this standard source tree, not within the package you initially receive.

Note: Do not modify any of these files except the configuration files and the PSP files.

2.1 Configuration Files

These files are in the directory `src/config`.

File	Description
<code>config_usbd_atmel_usbhs.c</code>	Defines the number of memory banks used for each endpoint.
<code>config_usbd_atmel_usbhs.h</code>	Contains all the configurable parameters. Configure these as required.

2.2 Source Code Files

These source code files are in the directory `src/usb-device/usb-drivers`. **These files should only be modified by HCC.**

File	Description
<code>usbd_atmel_usbhs.c</code>	Source code.
<code>usbd_atmel_usbhs_regs.h</code>	Register definitions.
<code>usbd_dev.h</code>	USB driver-specific header file.

2.3 Version File

The file `src/version/ver_usbd_atmel_usbhs.h` contains the version number of this module. This version number is checked by all modules that use this module to ensure system consistency over upgrades.

2.4 Platform Support Package (PSP) Files

These files are in the directory **src/psp/target**. They provide functions and elements the core code may need to use, depending on the hardware.

Note:

- These are PSP implementations for the specific micro-controller and development board; you may need to modify these to work with a different micro-controller and/or board. See [PSP Porting](#) for details.
- In the package these files are offset to avoid overwriting an existing implementation. Copy them to the root **hcc** directory for use.

File	Description
include/same70_regs.h	Register values.
usbd_atmel_usbhs/psp_usbd_atmel_usbhs.c	Functions source code.
usbd_atmel_usbhs/psp_usbd_atmel_usbhs.h	Sets the USBHS registers' base address.

The PSP also has a version file, **src/version/ver_psp_usbd_atmel_usbhs.h**.

3 Configuration Options

These are controlled by two files in the directory **src/config**.

- [config_usbd_atmel_usbhs.h](#)
- [config_usbd_atmel_usbhs.c](#)

3.1 config_usbd_atmel_usbhs.h

Set the following system configuration options in the file **src/config/config_usbd_atmel_usbhs.h**. This section lists the available options and their default values.

NO_OF_HW_EP

The number of hardware endpoints; the maximum is 10. The default is 10.

GET_N_BANKS(ep)

The number of banks a specific endpoint shall use. EP0 always uses 1 bank.

The default is $(ep < 1 ? 0 : (atmel_usbhs_ep_bank_cfg[(ep) - 1] - 1))$

ATMEL_DEVICE_ISR

The device interrupt. The default is `ISR_ID(USBD_ISR, DEVICE_ISR)`

ATMEL_DEVICE_INT_PRIO

The interrupt priority. The default is `configLIBRARY_LOWEST_INTERRUPT_PRIORITY`

ATMEL_EPT_DIR_IN

The endpoint direction. Keep the default of 1 to make this input.

ATMEL_EPT_DIR_OUT

The endpoint direction. Set this to of 1 to make this output. The default is 0.

ATMEL_ENABLE_LOW_POWER_MODE

Set this to 1 to enable low power mode. The default is 0.

ATMEL_FORCE_FS

Set this to 1 to force Full Speed operation (for testing). The default is 0.

3.2 config_usbd_atmel_usbhs.c

This file defines the number of memory banks used for each endpoint. For example, for an ATSAME70 the minimum is 1 and the maximum is 3, but only for EP1 and EP2. Please refer to the product datasheet for details for your device, or contact HCC support for help

```
const uint8_t atmel_usbhs_ep_bank_cfg[NO_OF_HW_EP - 1] =
{
  3 /* EP1 */
  , 3 /* EP2 */
  , 2 /* EP3 */
  , 1 /* EP4 */
  , 1 /* EP5 */
  , 1 /* EP6 */
  , 1 /* EP7 */
  , 1 /* EP8 */
  , 1 /* EP9 */
};
```

4 Integration

This section specifies the elements of this package that need porting, depending on the target environment.

4.1 OS Abstraction Layer

All HCC modules use the OS Abstraction Layer (OAL) that allows the module to run seamlessly with a wide variety of RTOSes, or without an RTOS.

This module requires the following OAL elements:

OAL Resource	Number Required
Tasks	0
Mutexes	0
Events	0
ISRs	1

4.2 PSP Porting

The Platform Support Package (PSP) is designed to hold all platform-specific functionality, either because it relies on specific features of a target system, or because this provides the most efficient or flexible solution for the developer. For full details of its functions and macros, see the *HCC Base Platform Support Package User Guide*.

The module makes use of the following standard PSP functions:

Function	Package	Element	Description
psp_memcpy()	psp_base	psp_string	Copies a block of memory. The result is a binary copy of the data.
psp_memset()	psp_base	psp_string	Sets the specified area of memory to the defined value.

The module makes use of the following standard PSP macro:

Macro	Package	Element	Description
PSP_RD_LE16	psp_base	psp_endianness	Reads a 16 bit value stored as little-endian from a memory location.

The module makes use of the following PSP functions, provided by the PSP to perform particular tasks. Their design makes it easy for you to port them to work with your hardware solution. The package includes samples in the **psp_usbd_atmel_usbhs.c** file.

Function	Description
usbd_hw_init()	Initializes the device.
usbd_hw_delete()	Deletes the device, releasing the associated resources.

These are described in the sections which follow.

usbd_hw_init

This function is provided by the PSP to initialize the device.

Note: Call this function first.

Format

```
int usbd_hw_init ( void )
```

Arguments

None.

Return Values

Return value	Description
USB_SUCCESS	Successful execution.
USB_ERROR	Operation failed.

usbd_hw_delete

This function is provided by the PSP to delete the device, releasing the associated resources.

Format

```
int usbd_hw_delete( void )
```

Arguments

None.

Return Values

Return value	Description
USB_SUCCESS	Successful execution.
USB_ERROR	Operation failed.