

USB Device Low Level Driver for ISP1582 User Guide

Version 1.10

For use with USB Device Low Level Driver for ISP1582
versions 1.10 and above

Date: 16-Jun-2017 16:03

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

System Overview	3
Introduction	3
Feature Check	4
Packages and Documents	5
Packages	5
Documents	5
Change History	6
Source File List	7
Source Code	7
Version File	7
Source Files in the Common Package	8
Configuration File	8
Source Code	8
Version File	8
Platform Support Package (PSP) Files	8
Configuration Options	10
Integration	12
OS Abstraction Layer	12
PSP Porting	12
psp_isp_init	14
psp_isp_r8	15
psp_isp_r16	16
psp_isp_r32	17
psp_isp_w8	18
psp_isp_w16	19
psp_isp_w32	20
psp_ispd_read_mem	21
psp_ispd_write_mem	22

1 System Overview

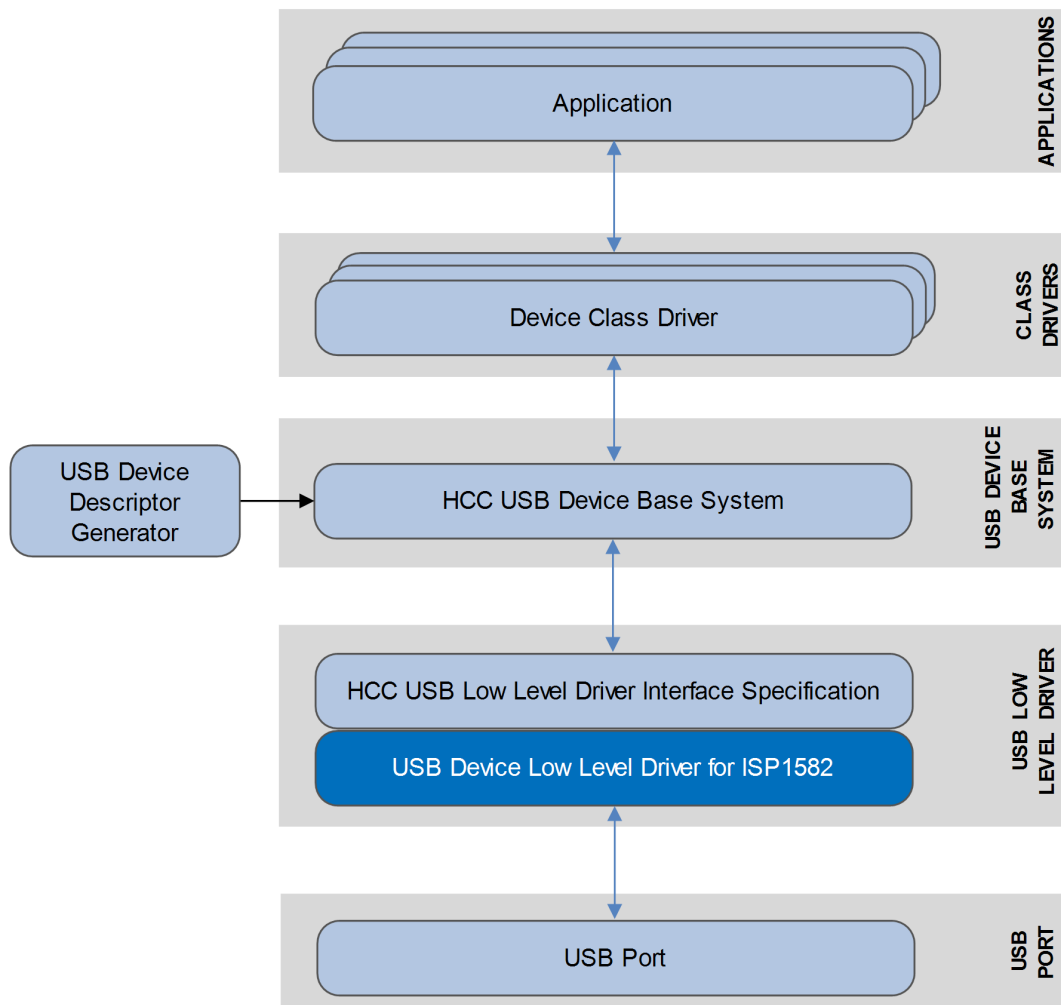
1.1 Introduction

This guide is for those who want to implement HCC Embedded's USB device stack with NXP Semiconductors' ISP1582 USB device low level driver.

The ISP1582 itself may be included within the SAF1761, a USB host controller, for which we provide a separate driver. The low level driver and host controller can be used together if required. This manual only covers the USB device functionality of the ISP1582. The SAF1761 is covered in the [HCC USB SAF1761 Host Controller User Guide](#).

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.
- Supports NXP Semiconductors' ISP1582 USB device low level drivers.
- Can work together with HCC's USB host controller for the SAF1761.
- Supports all USB transfer types: control, bulk, interrupt, and isochronous.

1.3 Packages and Documents

Packages

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

Package	Description
<code>hcc_base_doc</code>	This contains the two guides that will help you get started.
<code>usbd_base</code>	The USB device base package. Its source code includes the USB Driver device core.
<code>usbd_drv_isp1582</code>	The ISP1582 device low level driver package described by this document.
<code>usbc_drv_isp1582_isp176x</code>	Common code used by both this low level driver and the SAF1761 USB host controller module.

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.

HCC USB Device Low Level Driver for ISP1582 User Guide

This is this document.

1.4 Change History

This section describes past changes to this manual.

- To view or download earlier manuals, see [Archive: USB Device Low Level Driver for ISP1582 User Guide](#).
- For the history of changes made to the package code itself, see [History: usbd_drv_isp1582](#).

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	2017-06-16	1.10	New <i>Change History</i> format.
1.00	2015-12-22	1.10	First 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.

2.1 Source Code

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

File	Description
isp1582_reg.h	Header file for register settings.
usbd_dev.h	USB driver-specific header file.
usbd_isp1582.c	Source file for ISP1582 code.

2.2 Version File

The file **src/version/ver_usbd_isp1582.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.

3 Source Files in the Common Package

This section describes the source code files included in the common package used by both the ISP1582 module and the ISP176x device low level driver. 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 file and PSP files.

3.1 Configuration File

The file `src/config/config_usbc_isp1582_isp176x.h` contains the common configurable parameters. Configure these as required. For details of these options, see [Configuration Options](#).

3.2 Source Code

The source code files are in the directory `src/usb-common/usb-drivers/isp_1582_isp176x`. **These files should only be modified by HCC.**

File	Description
<code>usbc_isp1582_isp176x.c</code>	Source file for common code.
<code>usbc_isp1582_isp176x.h</code>	Header file for common public functions.

3.3 Version File

The file `src/version/ver_usbh_isp1582_isp176x.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.

3.4 Platform Support Package (PSP) Files

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

Note: These are PSP implementations for the specific microcontroller and development board; you may need to modify these to work with a different microcontroller and/or board. See [PSP Porting](#) for details.

File	Description
psp_isp1582_isp176x.c	Functions source code.
psp_isp1582_isp176x.h	Functions header file.

4 Configuration Options

Set the following system configuration options in the ISP176x/ISP1582 common package's file **src/config/config_usbc_isp1582_isp176x.h**. This section lists the available configuration options and their default values.

Note: For full details of these options, refer to the manufacturer's manual for the device.

ISP_USE_HOST

Set this to 0.

ISP_USE_DEVICE

Keep the default of 1 to enable the device (if one is available).

ISP_CHIP_VERSION

Ignore this option; keep the default of 3.

ISP_CHIP_VER_SAF

Ignore this option; keep the default of 0.

ISP_32BIT_IF

This specifies how the device is interfaced:

- 0 = 16 bit mode (the default).
- 1 = in 32 bit mode.

ISP_INT_LEVEL

The interrupt trigger type:

- 0 - level triggered.
- 1 - edge triggered (the default).

ISP_INT_POL

The interrupt polarity:

- 0 - active low (the default).
- 1 - active high.

ISP_ISR_ID

The ISR ID. The default is (`ISR_ID_XINT1`).

ISP_INT_Prio

The interrupt priority. The default is 0.

5 Integration

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

5.1 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	1
ISRs	1

5.2 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.

The host controller makes use of the following functions provided by the common ISP device package's `usbc_isp1582_isp176x.h` file.

Function	Description
<code>usbc_isp_init()</code>	Initializes the device.
<code>usbc_isp_delete()</code>	Deletes the device.
<code>usbc_isp_isr_install()</code>	Installs the ISR.
<code>usbc_isp_isr_enable()</code>	Enables the ISR.
<code>usbc_isp_isr_disable()</code>	Disables the ISR.
<code>usbc_isp_isr_delete()</code>	Deletes the ISR.
<code>usbc_isp_lock()</code>	Locks the device.
<code>usbc_isp_unlock()</code>	Unlocks the device.

The host controller makes use of the following functions provided by the common PSP template file `psp_isp1582_isp176x.h` file. These are target-specific.

Function	Description
<code>psp_isp_init()</code>	Initializes the device.
<code>psp_isp_r8()</code>	Reads an 8 bit ISP address.
<code>psp_isp_r16()</code>	Reads a 16 bit ISP address.
<code>psp_isp_r32()</code>	Reads a 32 bit ISP address.
<code>psp_isp_w8()</code>	Writes an 8 bit ISP address.
<code>psp_isp_w16()</code>	Writes a 16 bit ISP address.
<code>psp_isp_w32()</code>	Writes a 32 bit ISP address.
<code>psp_ispd_read_mem()</code>	Reads from the host memory area.
<code>psp_ispd_write_mem()</code>	Writes to the host memory area.

These functions are described in the following sections.

Note: HCC can provide samples for different configurations; contact support@hcc-embedded.com.

psp_isp_init

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

This enables the clocks, GPIO pin, external memory interface, and so on.

Format

```
int psp_isp_init ( void )
```

Arguments

None.

Return Values

Return value	Description
0	Successful execution.
Else	Operation failed.

psp_isp_r8

This function is provided by the PSP to read an 8 bit value from the specified address.

This macro is configurable for 16 and 32 bit addressing, based on the setting of [ISP_32BIT_IF](#).

Format

```
psp_isp_r8 ( d, a )
```

Arguments

Parameter	Description	Type
d	The destination; where to put the data read.	uint16_t
a	The address; where to read the data from.	uint16_t

psp_isp_r16

This function is provided by the PSP to read a 16 bit value from the specified address.

This macro is configurable for 16 and 32 bit addressing, based on the setting of [ISP_32BIT_IF](#).

Format

```
psp_isp_r16 ( d, a )
```

Arguments

Parameter	Description	Type
d	The destination; where to put the data read.	uint16_t
a	The address; where to read the data from.	uint16_t

psp_isp_r32

This function is provided by the PSP to read a 32 bit value from the specified address.

This macro is configurable for 16 and 32 bit addressing, based on the setting of [ISP_32BIT_IF](#).

Format

```
psp_isp_r32 ( d, a )
```

Arguments

Parameter	Description	Type
d	The destination; where to put the data read.	uint32_t
a	The address; where to read the data from.	uint32_t

psp_isp_w8

This function is provided by the PSP to write an 8 bit value.

Format

```
psp_isp_w8 ( a, v )
```

Arguments

Parameter	Description	Type
a	Where to write the value.	uint16_t
v	The value to write.	uint16_t

psp_isp_w16

This function is provided by the PSP to write a 16 bit value.

Format

```
psp_isp_w16 ( a, v )
```

Arguments

Parameter	Description	Type
a	Where to write the value.	uint16_t
v	The value to write.	uint16_t

psp_isp_w32

This function is provided by the PSP to write a 32 bit value.

Format

```
psp_isp_w32 ( a, v )
```

Arguments

Parameter	Description	Type
a	Where to write the value.	uint32_t
v	The value to write.	uint32_t

psp_ispd_read_mem

This function is provided by the PSP to read from the device memory area.

Format

```
void psp_ispd_read_mem (  
    uint8_t *   p_dst,  
    uint32_t   size )
```

Arguments

Parameter	Description	Type
p_dst	Where to put the data read.	uint8_t *
size	The number of bytes to read.	uint32_t

Return Values

None.

psp_ispd_write_mem

This function is provided by the PSP to write to the device memory area.

Format

```
void psp_ispd_write_mem (
    uint8_t *   p_src,
    uint32_t    size )
```

Arguments

Parameter	Description	Type
p_src	A pointer to the data to write.	uint8_t *
size	The number of bytes to write.	uint32_t

Return Values

None.