USB ISP1161 Host Controller User Guide

Version 1.10

For use with USBH ISP1161 Host Controller versions 2.01 and above

Date: 19-Jun-2017 17:03

Table of Contents

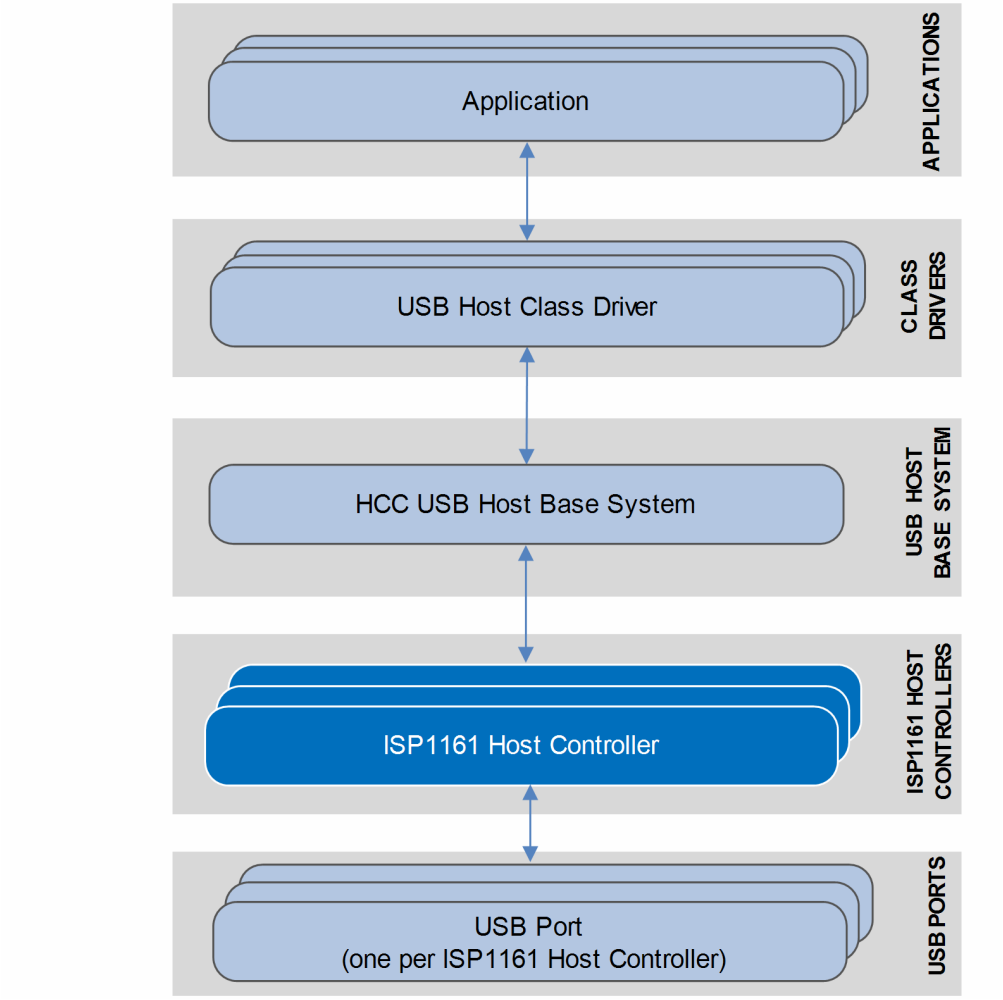
1 System Overview

1.1 Introduction

This guide is for those who want to implement HCC Embedded's USB host stack with NXP Semiconductors' ISP1161 and ISP1161A1 USB host controllers.

The module provides a high speed USB 2.0 host controller which provides both full and low speed USB functions. The controller can handle all USB transfer types and, in conjunction with the USB host stack, can be used with any USB class driver.

The position of the host controller within the USB stack is shown below:



1.2 Feature Check

The main features of the host controller are the following:

- Conforms to the HCC Advanced Embedded Framework.
- Designed for integration with both RTOS and non-RTOS based systems.
- Integrated with the HCC USB Host stack and all its class drivers.
- Supports NXP Semiconductor ISP1161 and ISP1161A1 USB host controllers.
- Supports multiple simultaneous ISP1161 host controllers, each with multiple devices attached.
- Spports all USB transfer types: control, bulk, interrupt, and isochronous.

1.3 Packages and Documents

Packages

The table below 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.
usbh_base	The USB host base package. This is the framework used by USB class drivers to communicate over USB using a specific USB host controller package.
usbh_drv_isp1161	The USB host controller package described by this document.

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 USB Host Base System User Guide

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

HCC USB ISP1161 Host Controller 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 ISP1161 Host Controller User Guide.
- For the history of changes made to the package code itself, see History: usbh_drv_isp1161.

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	2017-06-19	2.01	New *Change History* format.
1.00	2015-12-22	2.00	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 except the configuration file and PSP files.

2.1 API Header File

The file **src/api/api_usbh_isp1161.h** is the only file that should be included by an application using this module. It declares the **usbh_isp_hc()** function. For details, see Starting the Host Controller.

2.2 Configuration File

The file **src/config/config_usbh_isp1161.h** contains all the configurable parameters. Configure these as required. For details, see Configuration Options.

2.3 Source Code

The source code files are in the directory **src/usb-host/usb-driver/isp1161**. **These files should only be modified by HCC**.

File	Description
isp1161.c	Source file for ISP1161 code.
isp1161.h	Header file for ISP1161 public functions.
isp1161_hc.c	Source file for the ISP1161 HC descriptor.
isp1161_hc.h	HC descriptor header file.
isp1161_hub.c	Source file for public hub functions.
isp1161_hub.h	Header file for public hub functions.
isp1161_reg.h	Header file for ISP1161 registers.
isp1161_rw.c	Source code for functions including read/write.
isp1161_rw.h	Header file for functions including read/write.

2.4 Version File

The file **src/version/ver_usbh_isp1161.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.

2.5 Platform Support Package (PSP) Files

These files are in the directory named **src/psp/target/usbh-isp-port**. These provide functions and elements the core code may need to use.

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

The files are as follows:

File	Description
isp_port.c	Source code of the **isp_hwinit()** function.
isp_port.h	Header file.

3 Configuration Options

Set the following system configuration options in the file **src/config/config_usbh_isp1161.h**.

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

ISP_MAX_PORTS

The maximum number of ports in ISP1161. The default is 2.

MAX_DEVICE

The maximum number of devices supported. The default is 2.

MAX_EP

The maximum number of endpoints; the total of bulk, isochronous, and interrupt endpoints. The default is 4.

ISP_MAX_TRANSFER_SIZE

The maximum number of bytes one endpoint can send with one TX/RX call. The default is 1023.

ISP_BUF_SIZE

The buffer size. The default is 4096.

ISP_ATL_BUF_SIZE

The ATL buffer size. The default is 4096.

ISP_ITL_BUF_SIZE

The ITL buffer size. The default is 0.

ISP_MAX_TRANSFERS

The default is 8.

ISP_HOST_ISR

The VIC number of EINT2 in LPC2292. The default is VIC_EINT0.

ISP_HOST_INT_PRIO

The interrupt priority. The default is 0.

4 Starting the Host Controller

This section shows how to start the host controller and describes the task created. It includes a code example.

4.1 usbh_isp_hc

This external interface function provides the host controller descriptor required by the **usbh_hc_init()** function.

Format

```
extern void * const  usbh_isp_hc
```

4.2 Host Controller Task

The host controller task handles all completed transfers. Callback requested for the transfer is executed from this task.

The task has the following attributes:

Attribute	Description
Entry point	*usbh_isp_transfer_task*
Priority	USBH_TRANSFER_TASK_PRIORITY
Stack size	USBH_ISP_TASK_STACK_SIZE. The default is 4096.

4.3 Code Example

This example shows how to initialize the host controller. Note the following:

- There is only one external interface function, **usbh_isp_hc()**. To link this host controller to the system, you call the **usbh_hc_init()** function with this function as a parameter.
- The last parameter in the **usbh_hc_init()** call is the number of the host controller.

```
void start_usb_host_stack ( void )

{
int  rc;
rc = hcc_mem_init();


    if ( rc == 0 )
    {
        rc = usbh_init();    /* Initialize USB host stack */
    }

    if ( rc == 0 )
    {
        /* Attach ISP1161 host controller */
        rc = usbh_hc_init( 0, usbh_isp_hc, 0 );
    }

    if ( rc == 0 )
    {
        rc = usbh_start();  /* Start USB host stack */
    }


    if ( rc == 0 )
    {
        rc = usbh_hc_start( 0 );  /* Start ISP1161 Host controller */
    }

.....

}
```

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	1
Mutexes	1
Events	1
ISRs	0

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 module makes use of the following standard PSP function:

Function	Package	Element	Description
psp_memset()	psp_base	psp_string	Sets the specified area of memory to the defined value.

The host controller makes use of the following function provided by the PSP template file **psp_isp1161.h**.

Function	Description
isp_hwinit()	Initializes the device.

These function is described in the following section.

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

isp_hwinit

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 isp_hwinit ( void )
```

Arguments

None.

Return Values

Return value	Description
0	Successful execution.
Else	Operation failed.