

# USB Device Low Level Driver for ISP1181 User Guide

Version 1.20

For use with USB Device Low Level Driver for ISP1181  
versions 1.00 and above

**Date:** 06-Apr-2018 11:44

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	4
Feature Check	5
Packages and Documents	6
Packages	6
Documents	6
Change History	7
Source File List	8
Configuration File	8
Source Code	8
Platform Support Package (PSP) Files	9
Configuration Options	10
Integration	11
OS Abstraction Layer	11
PSP Porting	11
isp_hwinit	12

# 1 System Overview

This chapter contains the fundamental information for this module.

The component sections are as follows:

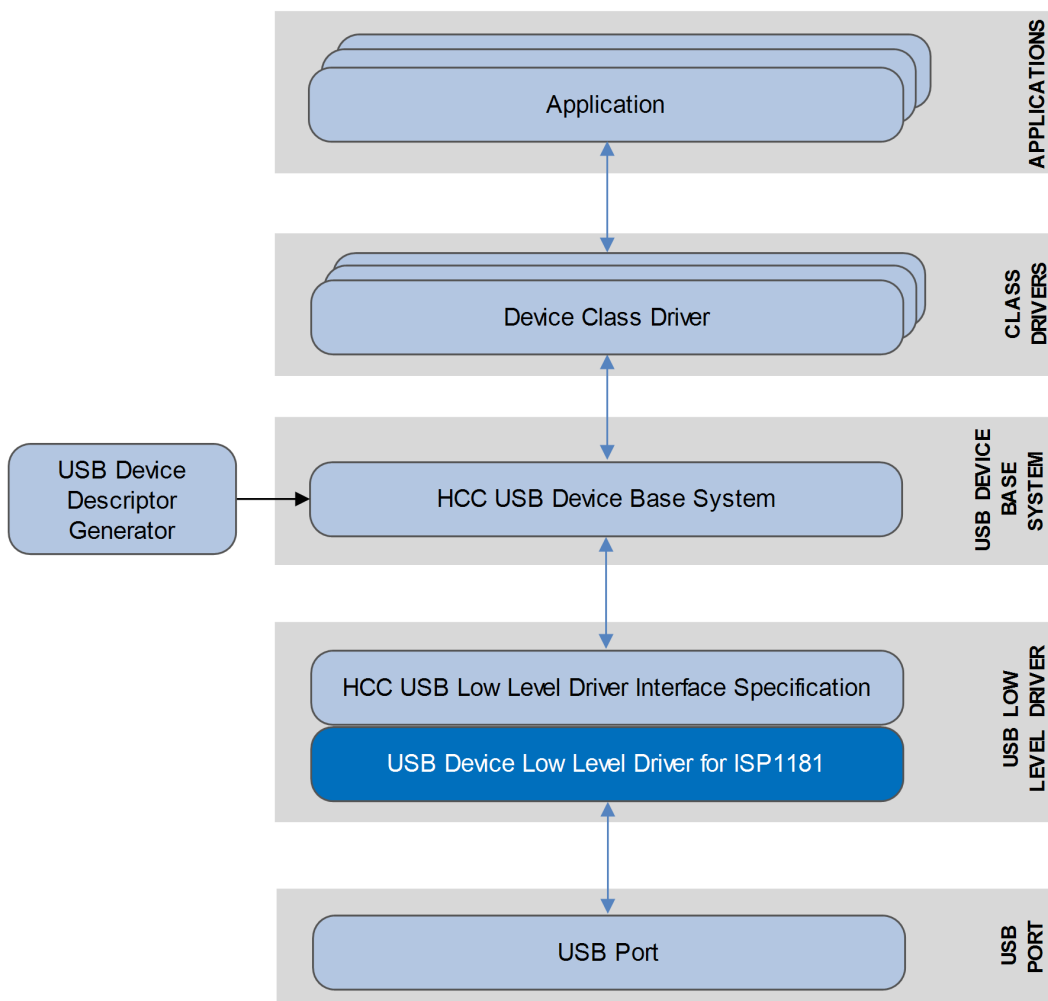
- [Introduction](#) – describes the main elements of the module.
- [Feature Check](#) – summarizes the main features of the module as bullet points.
- [Packages and Documents](#) – the *Packages* section lists the packages that you need in order to use this module. The *Documents* section lists the relevant 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 implement HCC Embedded's USB device stack with NXP Semiconductors' ISP1181 USB device low level driver.

The ISP1181 may be included within the ISP1362, 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 operation of the ISP1181. The ISP1362 is covered in the [HCC USB ISP1362 Host Controller User Guide](#).

This HCC driver can handle all USB transfer types and, in conjunction with the USB device stack, can be used with any USB device class driver. The 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' ISP1181 USB device low level drivers.
- Can work together with HCC's USB host controller for the ISP1362.
- 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_isp1181</code>	The ISP1181 device low level driver 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 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 ISP1181 User Guide**

This is this document.

## 1.4 Change History

---

This section describes past changes to this manual.

- To view or download manuals, see [USB Device PDFs](#).
- No changes have been made to the package code itself since its initial release.

The current version of this manual is 1.20. The full list of versions is as follows:

Manual version	Date	Software version	Reason for change
1.20	2018-04-06	1.00	Corrected <i>OS Abstraction Layer</i> table: removed event.
1.10	2017-06-16	1.00	New <i>Change History</i> format.
1.00	2015-12-23	1.00	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 Configuration File

The file `src/config/config_usbh_isp1181.h` contains all the configurable parameters. Configure these as required. For details of these options, see [Configuration Options](#).

### 2.2 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
<code>isp1181_reg.h</code>	Header file for register settings.
<code>isp1181_rw.c</code>	Source code for read/write functions.
<code>isp1181_rw.h</code>	Header file for read/write functions.
<code>usbd_dev.h</code>	USB driver-specific header file.
<code>usbd_isp1181.c</code>	Source file for ISP1181 code.



## 2.3 Platform Support Package (PSP) Files

---

These files are in directories under **src/psp/target/usb-isp-port**. 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
<b>isp_port.c</b>	Function source code.
<b>isp_port.h</b>	Function header file.

## 3 Configuration Options

Set the following system configuration options in the file `src/config/config_usbh_isp1181.h`.

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

### **ISP\_DEVICE\_ISR**

The VIC number of EINT2 in LPC2292. The default is VIC\_EINT2.

### **ISP\_DEVICE\_INT\_PRIO**

The priority of the ISP device interrupt. The default is 2.

## 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.

The module makes use of the following function provided by the PSP template file **isp\_port.h** file. This is target-specific.

Function	Description
<b>isp_hwinit()</b>	Initializes the device.

This function is described in the following section.

**Note:** HCC can provide samples for different configurations; contact [support@hcc-embedded.com](mailto: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.