# USB Device Low Level Driver for MSP430 User Guide

Version 1.00

For use with USBD Low Level Driver for MSP430 versions 1.07 and above

**Date:** 16-Mar-2018 10:31

# Table of Contents

# 1 System Overview

This chapter contains the fundamental information for this module.

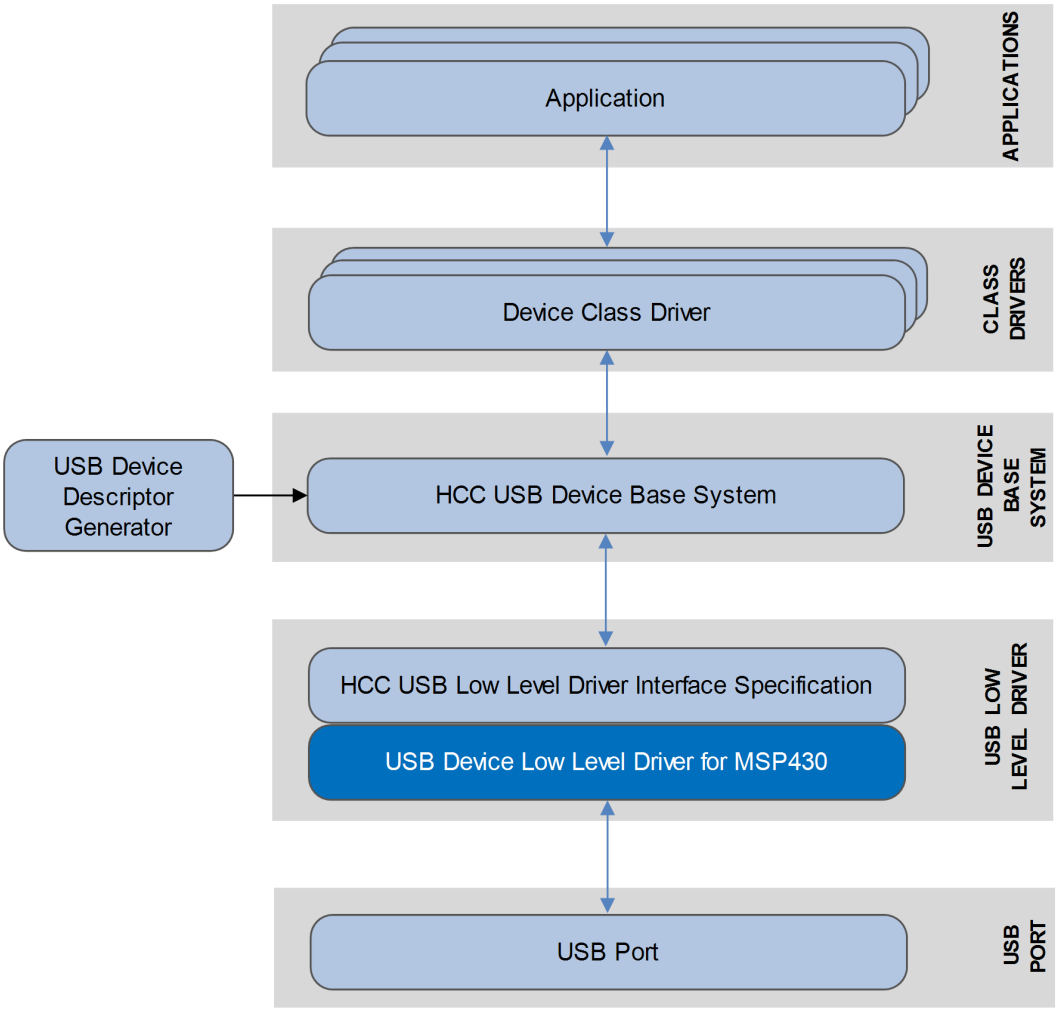
The component sections are as follows:

- Introduction – describes the main elements of the module.
- Feature Check – summarizes the main features of the module as bullet points.
- Device Description - presents a table summarizing some properties of these devices.
- 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 MSP430 module with HCC's USB device stack. This module provides a USB device driver for MSP430 microcontrollers from Texas Instruments Incorporated (the MSP430x1xx, MSP430x4xx, MSP430x552x, and MSP430F6638 families). 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 MSP430 microcontrollers from Texas Instruments Incorporated (MSP430x1xx, MSP430x4xx, MSP430x552x, and MSP430F6638 families).
- Supports all USB transfer types: control, bulk, interrupt, and isochronous.

## 1.3 Device Description

The driver supports a range of products. This table summarizes the properties of these devices.

The character that follows the "430" is usually "F" for Flash memory but other types are possible, including these:

- C - Masked ROM.
- FR - Ferroelectric RAM.
- L - RAM-only.

| Family | ROM | RAM | Note |
|---|---|---|---|
| MSP430x1xx | 1 - 16KB | 128B - 10KB | Basic device. |
| MSP430x4xx | 4 - 120KB | 256B - 8KB | Has integrated LCD controller, larger and more capable. |
| MSP430x552x | 4KB | 256KB | Has power management module to optimize power consumption and for integrated USB. |
| MSP430F6638 | 8KB | 256KB | Has power management module as above. |

### MSP430F5529 Devices

The following limitations apply to this type of device:

- The number of available endpoints on this device is 8 inclusive control endpoints.
- Only endpoint 0 can be configured to work as a control endpoint.
- Endpoints 1-7 can only be configured for INTERRUPT, BULK IN, and BULK OUT transfers.
- The packet size of the control endpoint is 8 bytes.
- The maximum packet size of other endpoints is 64 bytes.

## 1.4 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. |
| **usbd_base** | The USB device base package. Its source code includes the USB Driver device core. |
| **usbd_drv_msp430** | The MSP430 low level driver package described by this document. |
| **oal_base** | The base OS Abstraction Layer (OAL) package. |

### 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 MSP430 User Guide**

This is this document.

## 1.5 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_msp430.

The current version of this manual is 1.00.

| Manual version | Date | Software version | Reason for change |
|---|---|---|---|
| 1.00 | 2018-03-16 | 1.07 | 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_usbd_msp430.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 |
|---|---|
| **readme.txt** | Information on the setup. |
| **usbd_dev.h** | USB driver-specific header file. |
| **usbd_msp430.c** | Source code. |

## 2.3 Version File

The file **src/version/ver_usbd_msp430.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**. These 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.

The files are as follows:

| File | Description |
|---|---|
| **target/include/psp_msp430_regs.h** | Register definitions. |
| **board/init/cstartup.s43** | S43 file. |
| **board/init/target.c** | Function source code. |
| **board/init/target.c** | Function header file. |

# 3 Configuration Options

Set the system configuration options in the file **src/config/config_usbd_msp430.h**. This section lists the available configuration options and their default values.

**USE_ONE_EP**

If desired, set this to 1 to keep the code smaller if only a HID device is used. The number of hardware endpoints is normally 8, but just 2 are used if this is set.

**USBD_ISR**

The ISR. The default is USB_UBM_VECTOR.

**USBD_IT_PRIO**

The priority. The default is 0.

**EP0_PACKET_SIZE**

The endpoint 0 packet size. The default is 0x08.

**EPx_MAX_PACKET_SIZE**

The endpoint maximum packet size. The default is 0x40.

**UNLOCK_CONFIG_REGISTERS**

Unlock USB configuration registers for writing and wait until done. The default is USBKEYPID = 0x9628.

**LOCK_CONFIG_REGISTERS**

Lock USB configuration registers. The default is USBKEYPID = 0x9600.

**USB_BUFFER_BASE**

The base addresses of the transmit and receive buffers. The default is 0x1C00.

**EP1_OUT_X_BUFFER_ADDRESS**

The Input Endpoint 1 X buffer base address. The default is 0x1C00.

**EP1_IN_X_BUFFER_ADDRESS**

The Input Endpoint 1 Y buffer base address. The default is 0x1C40.

**EP2_OUT_X_BUFFER_ADDRESS**

The Output Endpoint 1 X buffer base address. The default is 0x1C80.

**EP2_IN_X_BUFFER_ADDRESS**

The Output Endpoint 1 Y buffer base address. The default is 0x1CC0.

**EP3_OUT_X_BUFFER_ADDRESS**

The Input Endpoint 1 X buffer base address. The default is 0x1D00.

**EP3_IN_X_BUFFER_ADDRESS**

The Input Endpoint 1 Y buffer base address. The default is 0x1D40.

**EP4_OUT_X_BUFFER_ADDRESS**

The Output Endpoint 1 X buffer base address. The default is 0x1D80.

**EP4_IN_X_BUFFER_ADDRESS**

The Output Endpoint 1 Y buffer base address. The default is 0x1DC0.

**EP5_OUT_X_BUFFER_ADDRESS**

The Input Endpoint 1 X buffer base address. The default is 0x1E00.

**EP5_IN_X_BUFFER_ADDRESS**

The Input Endpoint 1 Y buffer base address. The default is 0x1E40.

**EP6_OUT_X_BUFFER_ADDRESS**

The Output Endpoint 1 X buffer base address. The default is 0x1E80.

**EP6_IN_X_BUFFER_ADDRESS**

The Output Endpoint 1 Y buffer base address. The default is 0x1EC0.

**EP7_OUT_X_BUFFER_ADDRESS**

The Input Endpoint 1 X buffer base address. The default is 0x1F00.

**EP7_IN_X_BUFFER_ADDRESS**

The Input Endpoint 1 Y buffer base address. The default is 0x1F40.

**EP_NAK**

The endpoint NAK. The default is 0x80.

# 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 |