Micriμm

Empowering Embedded Systems

μC/USB-Host

V3.20.01

Release Notes

www.micrium.com

Revision History

Version	**Date**	**Description**
V3.20.01	2011 February	Improvements, corrections added.
V3.20	2009 September	New features, improvements, changes, corrections and limitations added.
V3.10a	2009 March	Corrections added.
V3.10	2008 December	New features, improvements, changes & corrections added.
V3.00	2008 August	Initial release.

Table of Contents

Required modules

Version 3.20.01

µC/CPU	Version 1.23 or later
µC/LIB	Version 1.30 or later

Version 3.20

µC/CPU	Version 1.23
µC/LIB	Version 1.30

Version 3.10a

µC/CPU	Version 1.22
µC/LIB	Version 1.29

Version 3.10

µC/CPU	Version 1.20
µC/LIB	Version 1.27

Version 3.00

µC/CPU	Version 1.16
µC/LIB	Version 1.25

New Features

Version 3.20.01

None.

Version 3.20

V3.20-001

The Host stack can be used without an operating system. The constant `USBH_CFG_OS_ENABLE` allows to configure the Host stack with or without OS.

Version 3.10a

None.

Version 3.10

V3.10-001

Added new functions and a new structure to handle the SCSI standard command, INQUIRY.

```
    USBH_SCSI_CMD_StdInquiry()
    USBH_MSC_StdInquiry()
    struct  msc_inquiry_info
```

V3.10-002

Use of the new function and the new structure to handle the SCSI standard command, INQUIRY.

```
    USBH_MSC_StdInquiry() called from MSFS_IoCtl()
    USBH_MSC_INQUIRY_INFO
```

V3.10-003

NEW class: CDC (Common Device Communication) which implements the Abstract Control Model (ACM). The CDC ACM allows to communicate with communicate with modem understanding the V.250 AT commands and to create virtual COM port

Version 3.00

Initial release.

Improvements

Version 3.20.01

None.

Version 3.20

V3.20-001

New constants:

- USBH_CFG_MAX_NBR_HC
- USBH_CFG_COMP_HC_TYPE_OHCI
- USBH_CFG_COMP_HC_TYPE_UHCI
- USBH_CFG_COMP_HC_TYPE
- USBH_CFG_OS_ENABLE
- USBH_CFG_HUB_EVENT_THREAD_PRIO,
- USBH_CFG_HUB_EVENT_THREAD_STK_SIZE,
- USBH_CFG_HUB_EVENT_QUEUE_SIZE
- USBH_CFG_PRINT_DEBUG

Constants removed:

- USBH_CFG_MAX_CURRENT
- USBH_CFG_ROOT_HUB_SPEED

V3.20-002

Modified two macros in OHCI driver to enhance the virtual to physical address translation when a processor with virtual memory is used. These macros are used in several functions.

```
#define  VIR2BUS(x)     USB_OS_VirToBus((x))
#define  BUS2VIR(x)    USB_OS_BusToVir((x))
```

V3.20-003

In the function `USBH_HostInit()`, a loop FOR has been added to enumerate the root hub of all the Host controllers registered during the initialization of the Host stack. Each root hub will be enumerated via the function `USBH_DevConn()`.

In function `USBH_HostSuspend()`and `USBH_HostResume()`, code modified for suspending all the root hubs of all the Host controllers registered into the stack.

In function `USBH_HostUninit()`, code modified to uninitialize all the root hubs from all the Host controllers registered into the stack.

V3.20-004

In usbh_core.h, Value of constant USBH_MAX_NBR_DEVICES changed from USBH_CFG_MAX_NBR_DEVICES + 1 to USBH_CFG_MAX_NBR_DEVICES + USBH_CFG_MAX_NBR_HC

Structure usbh_host modified:

V3.10a

```
struct  usbh_host {
USBH_HOST_STATE   HostState;
CPU_INT32U        HostBaseAddr;
CPU_INT32U        UsbMaxPower;
USBH_HOST_DRV    *DrvPtr;
void             *DataPtr;
USBH_DEV          DevList[USBH_MAX_NBR_DEVICES];
USB_POOL          DevPool;
USB_POOL_ENTRY    DevPoolEntry[USBH_MAX_NBR_DEVICES];
void             *AsyncQueueItems[USBH_CFG_ASYNC_QUEUE_SIZE];
USB_HQUEUE        AsyncQueue;
USB_HTHREAD       HAsyncThread;
USBH_DEV         *DevRootHubPtr;
USB_HMUTEX        HCDMutex;
};
```

V3.20

```
struct  usbh_host {
    USBH_HOST_STATE   HostState;
    CPU_INT32U        UsbMaxPower
    USBH_HOST_DRV    *DrvPtr;
    void             *DataPtr[USBH_CFG_MAX_NBR_HC];
    USBH_DEV          DevList[USBH_MAX_NBR_DEVICES];
    USB_POOL          DevPool;
    USB_POOL_ENTRY    DevPoolEntry[USBH_MAX_NBR_DEVICES];
    void             *AsyncQueueItems[USBH_CFG_ASYNC_QUEUE_SIZE];
    USB_HQUEUE        AsyncQueue;
    USB_HTHREAD       HAsyncThread;
    USBH_DEV         *DevRootHubPtr[USBH_CFG_MAX_NBR_HC];
    USB_HMUTEX        HCDMutex;
    CPU_INT08U        NbrHostCtrlr;
};
```

Functions removed:

- USBH_IsocTxAsync()
- USBH_IsocRxAsync()
- USBH_IfaceSet()
- USBH_HostGetFrmNbr()

Added function USBH_HostOTGPortSuspend()

V3.20-005

New error constants:

- USBH_ERR_DEV_LOW_SPEED,
- USBH_ERR_MAX_EHC_REACHED,
- USBH_ERR_MAX_OHC_REACHED,
- USBH_ERR_UNSUPPORTED_COMP_HC

V3.20-006

New functions:

- USBH_HUB_RefAdd()
- USBH_HUB_RefRel()
- USBH_HUB_EventThread()
- USBH_HUB_EvtPoll()
- USBH_HUB_PortSuspend()
- USBH_HUB_PortSuspendSet()

Creation of a new task USBH_HUB_EventThread():
In an OS context, this new task handles the hub events (connection, disconnection, reset and power) and root hub. Every 100 ms, the Hub thread will check for new events on all the ports of all the hubs.

Version 3.10a

None.

Version 3.10

V3.10-001

In OHCI driver, array OHCI_RHManufacturer[] initialized with the string MANUFACTURER.

V3.10-002

Function USBH_ClassDrvConn():

```
pcfg       = USBH_DevGetCfg(pdev, 0);
```
became
```
pcfg = USBH_DevGetCfg(pdev, (pdev->SelCfg - 1));
```

The function USBH_ClassDrvConn() attempts to find a class driver matching the device descriptor. If no class driver matches the device descriptor, it will attempt to find a class driver for each interface present in the active configuration. The field SelCfg is set with the number of configuration the Host has activated during the device configuration. Before the configuration #1 was always retrieved regardless the configuration number set by the Host with the SET_CONFIGURATION.

V3.10-003

In the MSC demo, USBHDemo_MSC_FileThread() modified to wait on semaphore. Each device connection will wake up the task and will occur a write-read-comparing file sequence.

V3.10-004

Function USBH_HID_IntrRxCB() modified for

- *bInterval* management if the Host controller is not a list-based controller or *bInterval* handled by the hardware controller
- avoiding error messages displayed while closing an endpoint because this is not an error.

V3.10-005

New function USBH_DevGetDesc().

Some functions have been removed:

- USBH_HostGetFrmNbr(),
- USBH_IsocTxAsync(),
- USBH_IsocRxAsync(),
- USBH_IfaceSet().

V3.10-006

New error constants:

- USBH_ERR_CFG_MAX_NBR_CDC_DEV
- USBH_ERR_CTRL_SIGNALS_NOT_SUPPORTED
- USBH_ERR_INVALID_LINESTATE_REQ
- USBH_ERR_FAIL for the CDC class
- USBH_ERR_STRING_DESCRIPTOR_NOT_PRESENT
- Host Controller IO Errors (-21 to -26)
  - USBH_ERR_IO_DATA_BUF
  - USBH_ERR_IO_BABBLE_DETECTED
  - USBH_ERR_IO_XACT_ERR
  - USBH_ERR_IO_MISSED_MICRO_FRM
  - USBH_ERR_IO_SPLIT_X_STATE
  - USBH_ERR_IO_PING_STATE

V3.10-007

New constant in usbh_cfg.h:

- USBH_CFG_PRINT_DEBUG

V3.10-008

In OHCI driver, value OHCI_MAX_NBR_DATA_BUFS changed from USBH_CFG_MAX_NBR_DEVICES to USBH_CFG_MAX_OPENED_EPS

Version 3.00

Initial release.

Changes

Version 3.20.01

None.

Version 3.20

None.

Version 3.10a

None.

Version 3.10

None.

Version 3.00

None.

Corrections

Version 3.20.01

V3.20.01-001

For the MSC class, in the storage layer for uC/FS v3.xx, `MSFS_StatusGet()` returned an inconsistent value to specify a successful or a failed initialization. The code has been modified to return the proper value.

Version 3.20

V3.20-001

In `usbh_inc.h`, the order of the header files inclusion has been modified to properly define the constants DEF_ENABLED and DEF_DISABLED:

V3.10a	V3.20

`#include  <usb_cfg.h>`
`#include  <cpu.h>`
`#include  <usb_def.h>`
	`#include  <usb_def.h>`
	`#include  <usb_cfg.h>`

Version 3.10a

V3.10a-001

Functions modified:

- `USBH_HUB_Suspend()`,
- `USBH_HUB_Resume()`,
- `USBH_HUB_Uninit()`,
- `USBH_HUB_EventReq()`,
- `USBH_HUB_EventProcess()`,
- `USBH_HUB_PortsInit()`,

The main reason for the changes was because the driver tried to dereference more `phub->DevPtrList` that were declared by `USBH_CFG_MAX_HUB_PORTS`. Thus causing exceptions on connect, disconnect, and other places (USBH_HUB_PortsInit, USBH_HUB_Uninit, etc).

In the function `USBH_HUB_EventReq()`:
The size of the request for the Hub Status Change was not following the USB standard.

```
err = USBH_IntrRxAsync((USBH_EP *) &phub->IntrEP,
                       (void    *) phub->HubIntrBuf,
                       (CPU_INT32U) 64,
                       (USBH_INTR_CMPL_FNCT)USBH_HUB_ISR,
                       (void    *) phub);
```

Replaced by

```
len = (phub->Desc.bNbrPorts / 8) + 1;
err = USBH_IntrRxAsync((USBH_EP *) &phub->IntrEP,
                       (void    *) phub->HubIntrBuf,
                       (CPU_INT32U         ) len,
                       (USBH_INTR_CMPL_FNCT)USBH_HUB_ISR,
                       (void              *) phub);
```

Hubs report only as many bits as there are ports on the hub, subject to the byte-granularity requirement (i.e., round up to the nearest byte) (See section 11.12.4, USB 2.0 spec).

Version 3.10

V3.10-001

Functions modified:

- USBH_HostCreate(),
- USBH_DevDisconn(),
- USBH_DevSetCfg(),
- USBH_EP_Open(),
- USBH_DfltEP_Open(),
- USBH_DevCfgRd().

In the function USBH_DevDescRd():
pdev->DfltEP.Desc.wMaxPacketSize = USB_MEM_RD_LE16U(&pdev->DevDesc[7]) replaced by
pdev->DfltEP.Desc.wMaxPacketSize = USB_MEM_RD_LE08U(&pdev->DevDesc[7])
because wMaxPacketSize for a Device Descriptor is 1 byte. A problem can occur during the device reconnection because wMaxPacketSize has a wrong value after reading memory.

In the function USBH_EP_Close():
pep->IsOpen = USB_FALSE; moved before USBH_URB_Abort(&pep->URB) to avoid error messages displayed while closing an endpoint.

In the function USBH_DevDisconn():
USBH_DevGetCfg() and pdev->HostPtr->UsbMaxPower removed.

In the function USBH_DevCfgRd ():
Local variable “bread” declared as CPU_INT16U instead of CPU_INT08U because USBH_GET_DESCRIPTOR() returns a two byte value.

Modified the return value of `USBH_EP_GetMaxPktSize`(). It returns a 2 byte because `wMaxPacketSize`” is always a 2 byte value.

Functions modified with several small changes:

- `USBH_DevSetCfg()`
- `USBH_EP_Open(),`
- `USBH_DfltEP_Open(),`
- `USBH_DevCfgRd().`

V3.10-002

In the function `USBH_DevConn()`, Added a condition for not asking Manufacturer and Product strings if not available in the USB device.

```
if(pdev->DevDesc[14] != 0) {
  USBH_PrintStrDesc(pdev,   (CPU_INT08U   *)"Manufacturer           :   ",   pdev->DevDesc[14]);
}

if(pdev->DevDesc[15] != 0) {
  USBH_PrintStrDesc(pdev,  (CPU_INT08U  *)"Product                    :  ",   pdev->DevDesc[15]);
}
```

Version 3.00

Initial release.

Known Problems

Version 3.20.01

None.

Version 3.20

None.

Version 3.10a

None.

Version 3.10

None.

Version 3.00

None.

Limitations

Version 3.20.01

See Limitations ‘V3.20-001’ to ‘V3.20-003’.

Version 3.20

V3.20-001

Isochronous transfer is not supported throughout the stack.

V3.20-002

The EHCI generic driver does not implement the Split Transaction Protocol used for LS and FS devices connected to a HS external hub.

V3.20-003

Multi-Host of same type (e.g. N OHCI controllers, N EHCI controllers) not supported.

Version 3.10a

None.

Version 3.10

None.

Version 3.00

Initial release.

Contacts

Micriμm
949 Crestview Circle
Weston, FL 33327
USA
+1 954 217 2036
+1 954 217 2037 (FAX)
e-mail: Licensing@Micrium.com
WEB: www.Micrium.com