



embeddedsolution
solutions

emUSB[®]

Versatile

Easy to use

ISO/ANSI-C source code

No royalties



USB stack

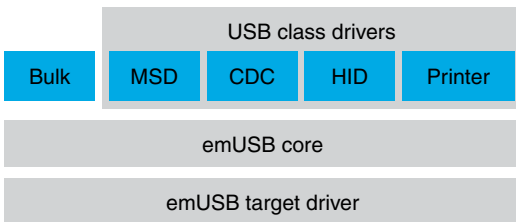
+++ 8 / 16 / 32 - bit +++



emUSB has been designed to work on any embedded system with a USB device controller. Ports for most common USB devices are available. It can be used with USB 1.1 or USB 2.0 devices.

emUSB Components

SEGGER offers a flexible USB device stack structure. The typical emUSB stack package consists of a target driver designed for your target hardware, the emUSB core and the Bulk, MSD, CDC or HID component or any combination thereof. The different available hardware drivers, the USB class drivers and the Bulk communication component are additional packages, which can be combined and ordered depending on the project requirements.



Components overview

Bulk Communication Component

emUSB-Bulk allows developers to quickly and painlessly develop software for an embedded device that communicates with a PC via USB. The communication is like a single, high speed, reliable channel (very similar to a TCP connection). It allows the PC to send and receive data with the embedded target. This permits usage of the full bandwidth of the USB bus.

MSD Component

emUSB-MSD converts your embedded target device into a USB mass storage device. Your target device can then be plugged into a USB host like a PC and accessed as an ordinary disk drive, without the need to develop a kernel mode driver for the host operating system. This is possible because the mass storage class is one of the standard device classes, defined by the USB Implementers Forum. Every major operating system on the market supports these device classes out of the box. Since every major OS already provides host drivers for USB mass storage devices, there is no need to implement your own. The target device will be recognized as a mass storage device and can be accessed directly.

Typical applications:

- Digital camera
- USB stick
- MP3 player
- DVD player
- Any target with a USB interface: Easy access to configuration and data files

CDC Component

emUSB-CDC converts the target device into a serial communication device. The host will recognize it as a virtual COM port (USB2COM). It allows the use of software which is not designed to be used with USB, such as a device datalogger or terminal program.

HID Component

The Human Interface Device class (HID) is an abstract USB class protocol defined by the USB Implementers Forum. This protocol was defined for the handling of devices which are used by humans to control the operation of computer systems. An installation of a custom-host USB driver is not necessary, because the USB human interface device class is standardized and every major OS already provides host drivers for it.

Typical applications:

- Keyboard
- Mouse and similar pointing devices
- Joystick
- Game pad
- Front-panel controls - for example, switches and buttons.

emUSB device also allows "Vendor specific HIDs". These are HID devices communicating with an application program. The host OS loads the same driver loaded for any "true HID" and automatically enumerate the device.

The application communicates with the particular device using API functions offered by the host. This allows an application program to communicate with the device without the need of loading a custom driver. The HID class is a good choice, if ease-of use is

Features

- ISO/ANSI C source code
- Supports USB 1.1 / 2.0 devices
- Full/High Speed support
- Bulk communication component with Windows kernel mode driver available
- MSD component available
- MSD - CDRom support
- MSD supports any type of storage incl. NAND
- CDC component available
- HID component available
- Printer component available
- Start/test applications supplied
- No royalties

important and high communication speed is not a requirement.

Typical applications:

- Bar-code reader
- Thermometer
- Voltmeter
- Low-speed JTAG emulator
- UPS (Uninterruptible power supply)

Printer Component

The USB class protocol for printers was defined for the handling of output devices like printers and plotters. emUSB printer receives data from the host and forwards the data to a parser. The printer component provides automatic error handling routines, if events occur like the device running out of paper. The USB protocol is completely hidden for the developer and he can concentrate on developing the parser.

emUSB Host

Segger's USB host software stack implements full USB host functionality, including external hub support, and optionally provides device class drivers. It enables developers to easily add USB host functionality to embedded systems.

The software stack complies with the USB v1.1 and USB v2.0 specifications. It supports all transfer modes (control, bulk, interrupt, isochronous) at low, full and high speed.

SEGGER offers ready-to-use trial versions for common evaluation boards



Öä^&Q•ä @ScãZíc] KD , Èã^&ç•ä @B È Æ Ð
V@P æ || -Ö!^æ || @PãZÖ!^æ || @Öä à !' ÈUç !ã • @^UÿF! ÄÖP ÈMS
V^KE | ÄÖFGJ Ä ! ! | €^WÖæK È | ÄÖFGJ Ä ! G JJ^ã • Ö äã^&ç • ä @B È