

# USBFS-DEV\_SS

USB 2.0 Full Speed Peripheral Device Software Stack



## MAJOR FEATURES

- ◆ Compliance with USB 1.1/2.0 standard
- ◆ Complete support for the Evatronix USBFS-DEV controller
- ◆ Optimized for 32 bit microprocessors
- ◆ Written in ANSI C
- ◆ CPU and OS independent
- ◆ Designed for applications of the majority of USB classes

## APPLICATIONS

- ◆ HID - mouse, volume control
- ◆ Vendor specific

## RELATED PRODUCTS

**USBFS-DEV** - USB 2.0 Full Speed device controller with a serial PHY interface and a generic/AMBA®/P/PCI interfaces for the CPU.

**USBFS-51\_AP** - a complete System-on-Chip solution that integrates the **R8051XC2** microcontroller with the **USBFS-DEV**, while the complementary **USBFS-51\_SS** software stack further facilitates application development and FPGA prototyping processes.

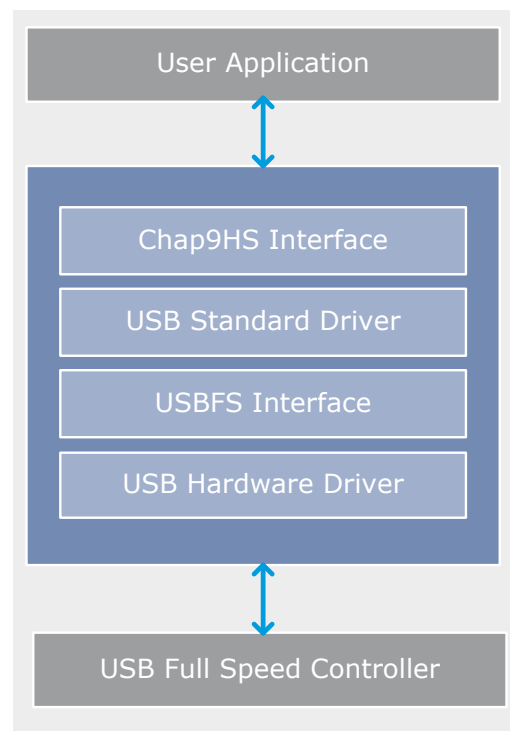
## OVERVIEW

The **USBFS-DEV\_SS** is a USB 1.1/2.0 compatible software stack for peripheral devices designed to take full advantage of a compliant microprocessor as well as the USB 2.0 Full Speed Controller.

The software stack provides the USB device with user-friendly Application Programming Interface (API), which allows development of applications without a detailed knowledge of the USB Full Speed controller hardware.

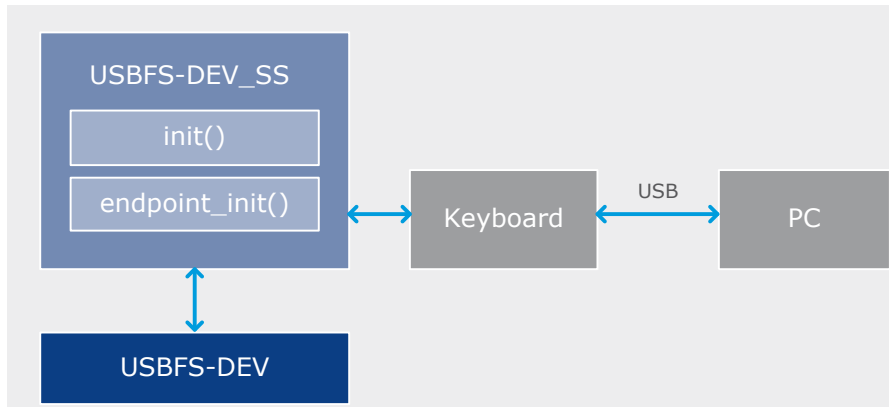
A rich set of attached applications reduces the learning curve and allows writing robust embedded software for USB-enabled Systems-on-Chips.

## BLOCK DIAGRAM



# USBFS-DEV\_SS

## EXAMPLE APPLICATION



The figure above illustrates an example application of the **USBFS-DEV\_SS**.

The **USBFS-DEV\_SS** uses two basic callback functions, `evx_usb_init_application()` and `config_callback()`, to configure device example functionality.

The basic task of the `evx_usb_init_application()` function is to register application specific descriptors. Another task of the `evx_usb_init_application()` function is the registration of a class specific requests handler.

Since **USBFS-DEV\_SS** realizes request service from hardware layer up to chapter 9 level, the user has to write his own class or vendor specific request service (placed above chapter 9 layer).

The `config_callback()` function is used mainly for endpoint initialization. When initialized, a built-in USBFS Interface allows endpoints to work in each of transmission types: bulk, control, isochronous or interrupt.

## BENEFITS

- ◆ The **USBFS-DEV\_SS** allows for a USB application development without detailed hardware knowledge, eg. only the DAC interface knowledge is required when using the Software Stack with USB Audio Speaker Class
- ◆ Efficient design
- ◆ Intuitive API

## STANDARD DELIVERABLES

- ◆ Software Stack C source code
- ◆ User documentation
- ◆ Example applications
- ◆ 30 days of technical support
- ◆ 90 days of warranty against defects

## DELIVERY OPTIONS

- ◆ One-year maintenance
- ◆ On-site support
- ◆ On-demand modifications



For more information on our IP portfolio visit [www.evatronix.com/ip](http://www.evatronix.com/ip)



**EVATRONIX SA**  
Przybyły 2, 43-300 Bielsko-Biała, Poland  
T: +48 33 499 59 15  
F: +48 33 499 59 18

info@evatronix.com  
www.evatronix.com/ip