Embedded USB made easy

With USB increasingly becoming the standard interface for embedded systems, Manuel Schreiner and Markus Vogel from Fujitsu Semiconductor Europe explain how development can be simplified using the company’s USB software library for the 16- and 32-bit microcontrollers.

The universal serial bus (USB) is increasingly becoming the standard interface for embedded systems of all kinds. In addition to the ability to connect portable storage media such as USB sticks to control devices such as a mouse or keyboard to the device, the connection to a PC is one of the most frequent uses. This connection is used for different purposes such as updating device firmware, exchanging data or simply setting specific operating and control parameters. USB can also be used as a general-purpose interface for servicing and replacing well-established serial interfaces such as RS232, PS/2 and even the parallel Centronics/IEEE-1284. This is because increasingly large volumes of data have to be transferred and modern PC systems and notebooks are simply no longer equipped with these interfaces and have a large number of USB ports instead. Other factors associated with the advance of USB are the high error correction rate, plug & play capability and the possibility of supplying devices with a stabilized voltage supply through the USB interface.

Technology

Developing USB-compliant firmware components usually requires the programmer to have an in-depth understanding of the USB specification and the hardware macros implemented on the chip. If no USB classes stored in the operating system are applied when the embedded system is used as a USB function (i.e. the slave connection to a PC), a specific driver has to be developed. This also requires an understanding of how to programme the PC operating system. Because most developers shy away from this expense, they usually resort to standard classes that have already been implemented, such as human interface devices (HID) or mass storage devices (MSD).

Various RS232-to-USB converter modules have also enjoyed considerable success on the market, but although these enable a USB connection to the PC, they only convert data to the standard RS232 format on the embedded side. On the PC side, a virtual COM port is simulated with the communication device class (CDC). The required driver is implemented in the Windows and Linux-based operating systems. However, this solution can only be used in a few cases and is limited by the performance of the serial RS232 protocol.

To be able to use the entire data bandwidth as well as the various USB transmission modes such as interrupt, bulk or isochronous transfer (depending on the application), it is essential that a full USB class code is implemented. Fujitsu provides project templates for developing firmware for its 16-bit microcontrollers (MB90300 series) and 32-bit microcontrollers (MB91656/655 series). These templates contain a USB function library which is suitable for the basic tasks of the USB such as procedures for logging-in and -off the bus (also known as enumeration) and setting up ‘endpoints’ for data transmission. Based on these functions, the library contains a sample implementation of a customer-specific USB vendor class. The implementation of typical standard classes such as HID or MSD is shown with demo examples. To help you get started, suitable USB development tools that help you create a USB device are needed. Those include the actual USB configuration (USB descriptor) and the communication module which controls the interface between the application and driver (USB class). The Fujitsu USB Assistant, which creates all of the files needed for USB as C-Code, is just such a tool.

Transparency

To make the USB class more transparent between the driver and user application, the configuration, as a USB class, creates a template for communication routines that are adapted to the corresponding configuration. The USB class mainly deals with data packets that are received or sent as interrupt-controlled block transfers and provides data for the user application. Finally, the transmission protocol is converted in the USB class and can be controlled by the user application through API, thereby enabling the embedded application to easily access the USB.

The universal serial bus (USB) is clearly becoming the standard interface for embedded systems of all kinds.

The universal serial bus (USB) is clearly becoming the standard interface for embedded systems of all kinds.

Southern Manufacturing

The UK’s largest and longest established Manufacturing Technology Exhibition

The UK’s largest and longest established Manufacturing Technology Exhibition

M

Over 500 national and international suppliers under one roof in
Farnborough this February at Southern Manufacturing & Electronics (Inc AutoAsia) 2011.

Over 500 national and international suppliers under one roof in
Farnborough this February at Southern Manufacturing & Electronics (Inc AutoAsia) 2011. See live demonstrations and new product launches of machine tools & tooling, electronics, factory & process automation, packaging & handling, labelling & marking, test & measurement, materials & adhesives, rapid prototyping, ICT, drives & controls and laboratory equipment.

Free industry seminar programme online at www.industry.co.uk

The exhibition is free to attend, free to park and easy to get to.

Doors open at 10am on Wednesday 16 February.

Pre-register online now for your free entry badge and show preview at www.industry.co.uk