

ES623 Networked Embedded Systems

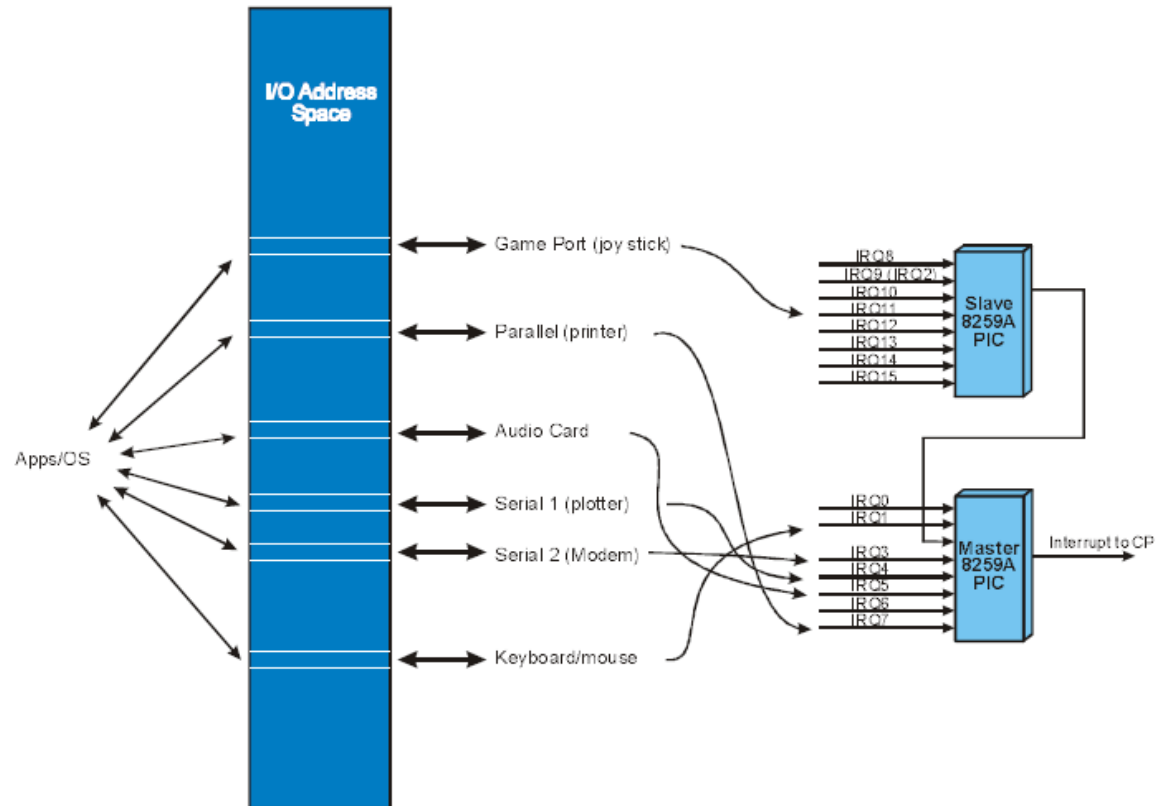


Universal Serial Bus

03rd May 2013

Universal Serial Bus (USB)

- USB emerged as a result of the difficulties associated with the cost, configuration, and attachment of peripheral devices in the personal computer environment.



USB

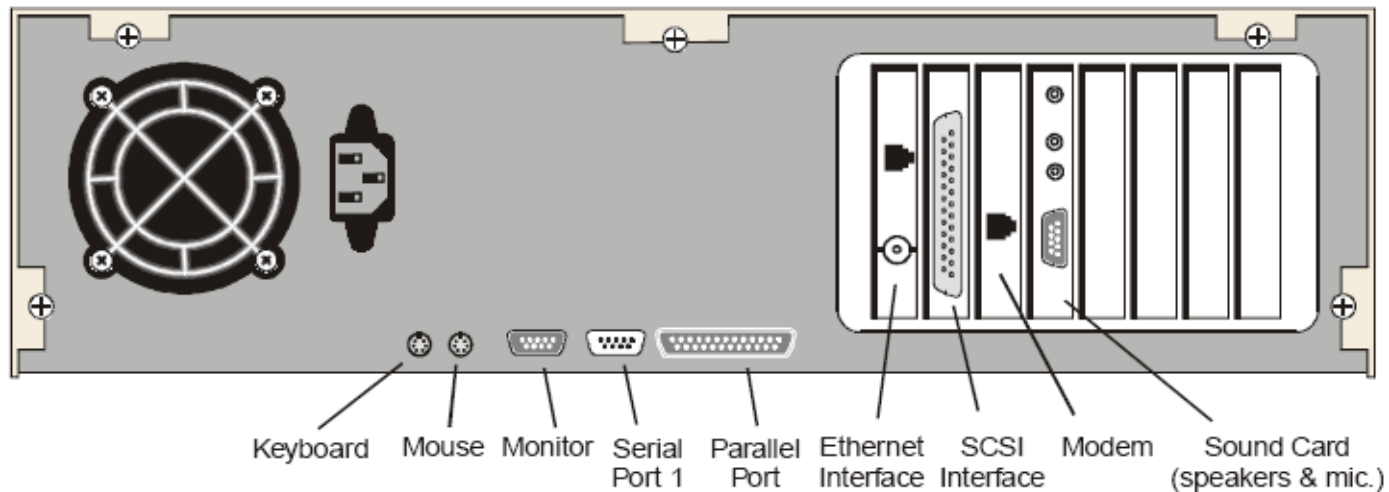
- I/O address conflicts are common in the PC environment.
- Peripheral devices require a block of I/O address locations to report status information and to issue commands to the device.
- Standard PC peripheral interfaces support the attachment of a single device.
- Only one peripheral device can be attached at any given time, the flexibility of such connections is minimized.



USB

End User Concerns

- Too many connector/cable types
- System must be shut down to attach most peripherals
- System must be restarted to install/load software
- Cost
- Dedicated cables are required for the mouse, keyboard, printer, external modem, Zip drive, plotter, etc., most of which are completely different



USB Paradigm

- a single connector type to connect any PC peripheral
- ability to attach many peripheral devices to the same connector
- a method of easing the system resource conflicts
- hot plug support
- automatic detection and configuration of peripheral devices
- low-cost solution for both system and peripheral implementations
- enhanced performance capability
- support for attaching new peripheral designs
- support for legacy hardware and software
- low-power implementation



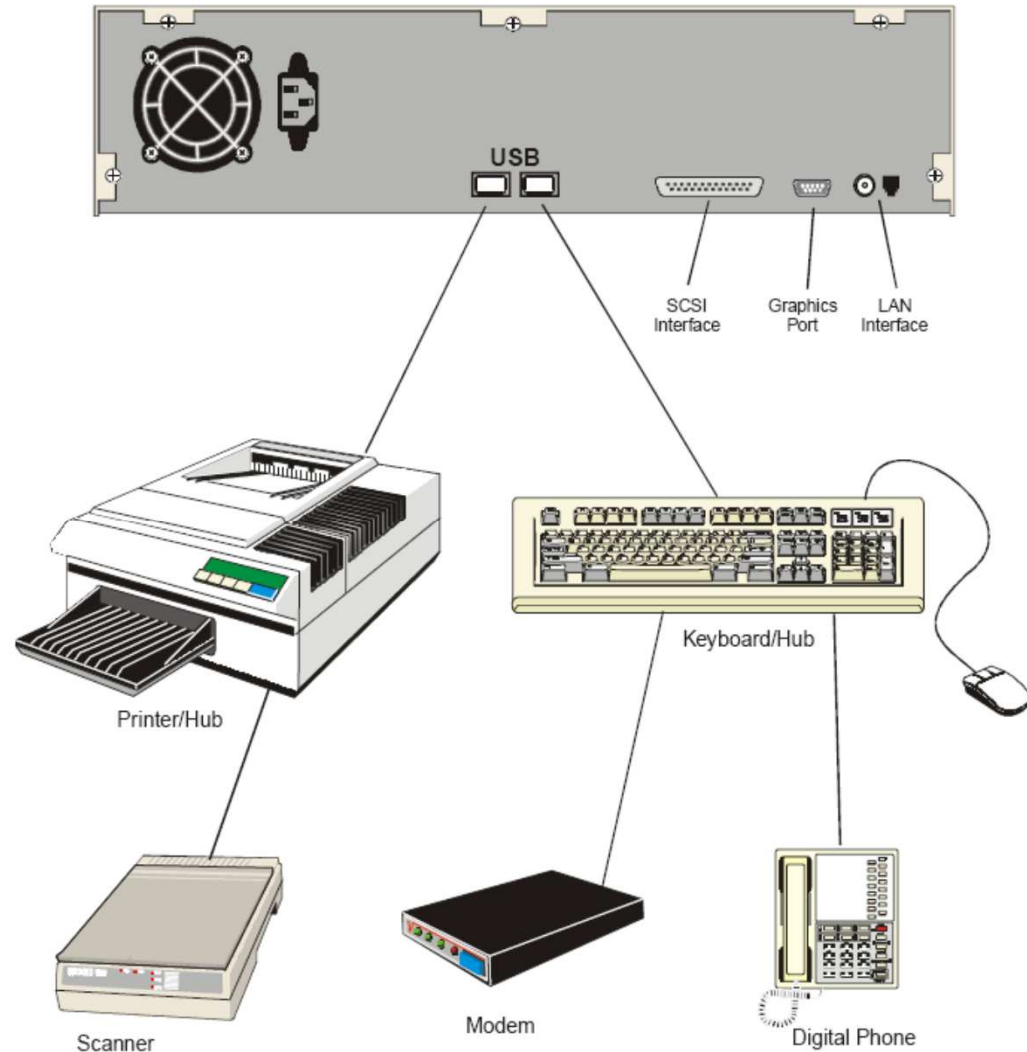
USB Performance

- USB supports three transmission rates:
 - 1.5Mb/s
 - 12Mb/s
 - 480Mb/s
- The 1.0 and 1.1 (1.x) versions of USB support only the 1.5 Mb/s and 12Mb/s speeds.
- 2.0 version of the USB specification defines a 480Mb/s rate that can support selected high-speed devices, and permits a larger number of low- or full-speed devices to operate on a single bus.



USB

- Hot Plug and Play Support
- Expandability
- Low Cost



USB Features

Feature	Description
Low Cost	The USB provides a low-cost solution for attaching peripheral devices to PCs.
Hot Pluggable	Device attachment is automatically detected by the USB and software automatically configures the device for immediate use, without user intervention.
Single Connector Type	The USB defines a single connector used to attach any USB device. Additional connectors can be added with USB hubs.
127 Devices	Supports the attachment of 127 devices per USB.
Low Speed, Full Speed, and High Speed Devices	The USB 2.0 supports three device speeds: 1.5Mb/s, 12Mb/s, and 480Mb/s.
Cable Power	Peripherals can be powered directly from the cable. 5.0vdc power is available from the cable. The current available can vary from 100ma - 500ma depending on the hub port.



How Transactions are generated

- USB 1.x system implementations generate USB transactions by fetching and executing a linked list of data structures from memory.
- Each transfer descriptor defines a USB transaction that software has requested and scheduled for the purpose of accessing a USB device.



How transactions are generated

- Each transfer descriptor contains information that describes a transaction to be performed.
- The primary information includes:
 - The USB device address
 - The type of transaction to be performed (read or write)
 - The transfer size
 - Speed of the transaction
 - The location of the memory data buffer



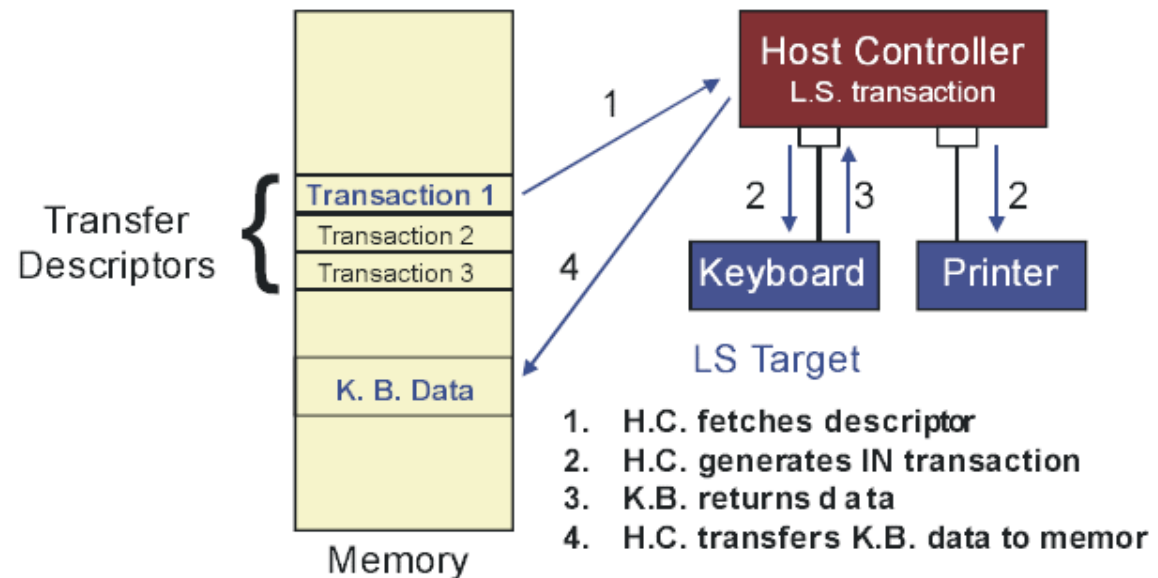
How transactions are generated

- The linked list of descriptors is sometimes called a **transaction list or frame list**.
- During a 1ms interval (called a **frame**), the host fetches and executes a series of descriptors.
- Consider a **keyboard being polled by software** to check if a key has been pressed.
- The direction of data flow in USB is specified with respect to the host.
- Since data is being read by the host, the transaction is termed an **IN transaction**.



How transactions are generated

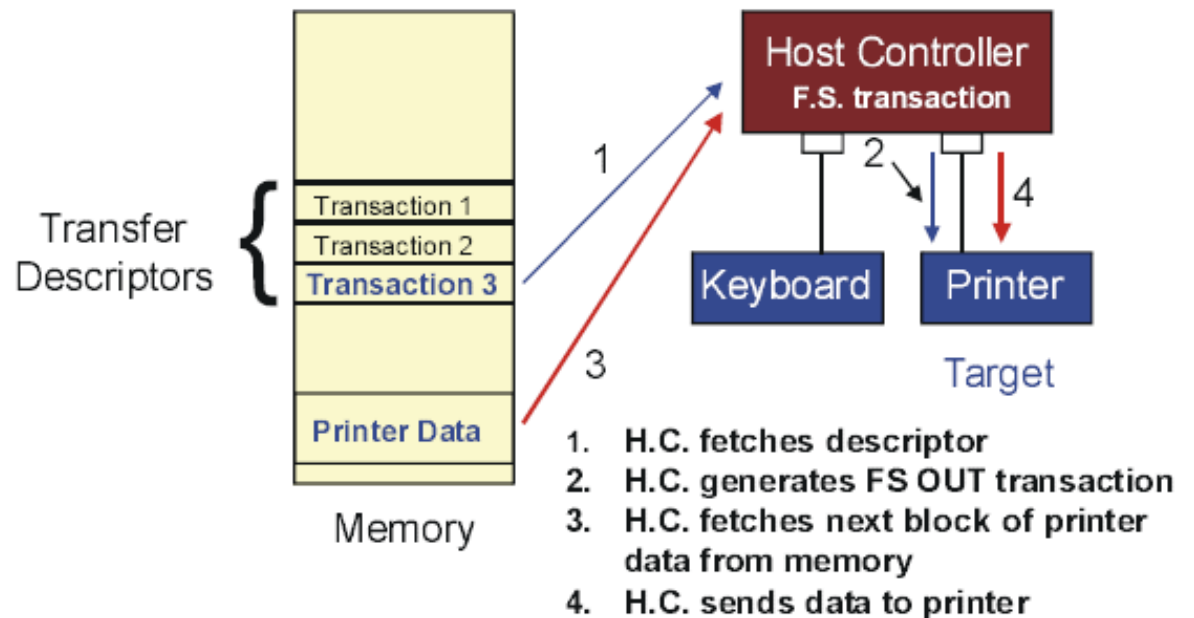
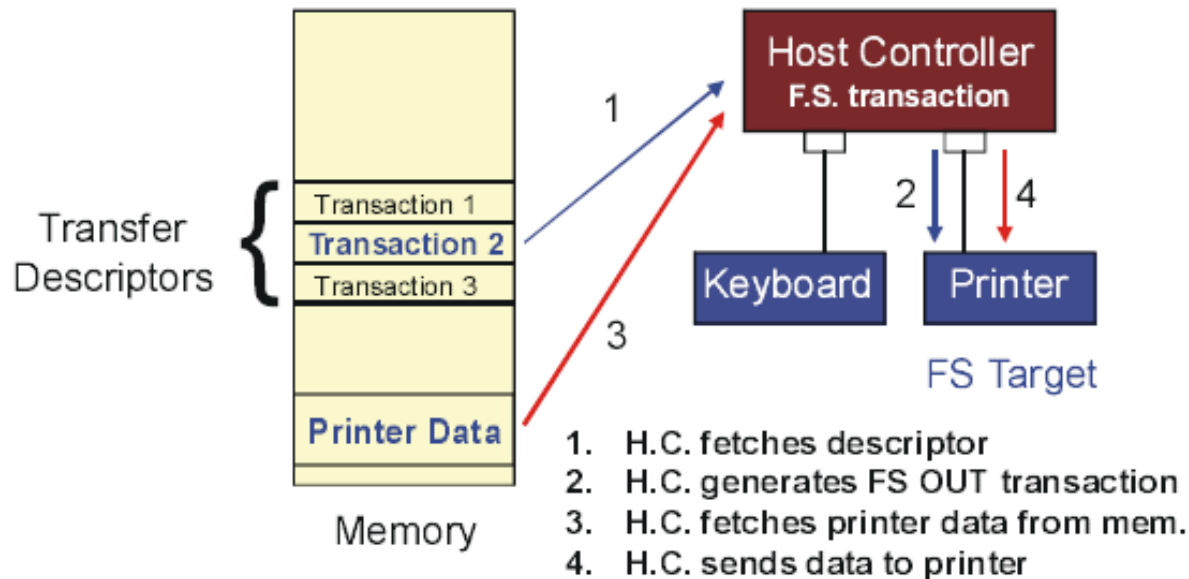
- Previously the keyboard's USB driver has requested that the keyboard be polled periodically to determine if the user has pressed a key.
- The driver also supplies a memory buffer location where the keyboard data is to be returned.
- This request has resulted in the host software creating a transfer descriptor in memory that describes the USB polling operation



How transactions are generated

- **Host controller fetches transaction 1 and decodes** the descriptor and executes the requested IN transaction.
- The keyboard **returns data to the host controller**, which in turn places the data into the keyboard data area in memory.
- The **pointer to the keyboard data buffer** is included within the transfer descriptor.
- The **keyboard software driver reads** the keyboard data buffer to acquire the data.
- The second and third descriptors define transactions that send data to the USB printer.
- The direction of data flow in this case is **OUT from the host**.





Hardware and Software involved

- All USB transactions are initiated by USB software.
- These accesses are typically originated by a USB device driver that wants to communicate with its device.
- The USB driver provides the interface between USB device driver and the USB host controller.
- This software is responsible for translating client requests into transactions that send information either to or from a target USB device.



Hardware and Software involved

- The primary hardware and software elements associated with a USB solution includes:
- USB Hardware
 - USB Host Controller/Root Hub
 - USB Hubs
 - USB Devices
- USB Software
 - USB Device Drivers
 - USB Driver
 - Host Controller Driver

