

UNIVERSAL SERIAL BUS

Bandaru Raja Sekhar (02307046)
Supervisor: Prof P.C.Pandey

Abstract

Universal Serial Bus is an interface for communicating external peripherals with a PC. In this report, various interfaces prevalent today and their comparisons with USB are brought out. Different types of USB controllers along with the chips available in market is discussed in detail. The USB hardware and software details along with USB products available in market are presented. The advantages both from the user and developer point of view are included. A brief discussion on different data transfers supported by USB and how the peripheral gets enumerated as soon as it is plugged into a port is also included.

Introduction

Universal Serial Bus (USB) is primarily a PC bus, which is designed so that simple and efficient way of communication is possible with many types of peripherals, without the frustrations and limitations of existing interfaces. In other interfaces, each peripheral depending on its speed has its own type of interface. Another major disadvantage while connecting a new peripheral, an Extended Industry Standard Architecture (EISA) card is to be connected as limited Peripheral Component Interconnect (PCI) slots exist on a mother board. To overcome all these problems and ease in peripheral connectivity, an organization called Universal Serial Bus –Implementers Forum [1] formed to do away with all the existing different varieties of interfaces and to come with an ideal interface which have the following features.

- Ease of use for PC peripheral expansion
- Low cost solution that supports different transfer rates from few kb/s to hundreds of Mb/s
- Full support for real-time data for voice, audio and video
- Plug and play etc.

A number of interfaces is prevalent today. Each peripheral depending on its transfer data uses suitable interface. In the present convention, low speed and interrupt devices use serial interface while high speed devices use parallel interface. The popular interface and comparison with USB has shown in Table 1. An interface with the above specified ideas was developed and named as Universal Serial Bus. The copyright on USB1.1 specifications [1] is assigned jointly to four corporations, all popular in PC hardware and software: COMPAQ, INTEL, MICROSOFT and NEC. All have agreed to make the specifications available for use by anyone without charge. As it gained widespread acceptance in industry, other three joined namely HEWLETT PACKARD, LUCENT TECHNOLOGIES and PHILIPS and the latest version USB2.0 [1] copyright is jointly owned by them. USB supports a wide range of data rate transfers and thus various peripherals can use USB as an interface. Table 2 gives an overview.

Table 1: A comparison of popular interfaces [3].

Interface	Format	Max no. of devices	Cable length (in feet)	Speed (in bits/s)	Typical use
RS-232	Asynchronous serial	2	50-100	20k	Modem, mouse
RS-485	Asynchronous serial	32 unit loads	4000	10M	DAS, Control System
SPI	Synchronous serial	8	10	2.1M	Micro-controller communications
IEEE-488(GPIB)	Parallel	15	60	8M	Instrumentation
Parallel printer port	Parallel	2 (or 8 with daisy chain)	10-30	8M	Printers, scanners
IEEE-1394a, IEEE-1394b	Serial	64	15	400M, 3.2Gb/s	Video
USB	Asynchronous serial	127	16 (or 96 with 5 hubs)	1.5M, 12M, 480M	Mouse, printers, Video

Table 2: USB application taxonomy

Performance	Applications	Attributes
Low speed Interactive devices 10-100kb/s	Keyboard, mouse, stylus	Lowest cost Ease of use Dynamic attach/detach
Full speed Phone, audio, compressed video 500kb/s-10Mb/s	Broadband, audio, microphone	Lower cost Ease of use Guaranteed B.W. Guaranteed latency
High speed Video, storage 25-480Mb/s	Video Storage Imaging Broadband	Low cost Multiple Peripherals Guaranteed B.W. Guaranteed latency

Inside USB

USB connects USB devices with host. USB physical interconnect is a tiered star topology and its architecture overview is shown in Fig.1. Due to timing constraints allowed for hub and cable propagation times, the maximum number of tiers allowed is seven including the root hub. A compound device occupies two tiers, and therefore cannot be enabled at tier level seven.

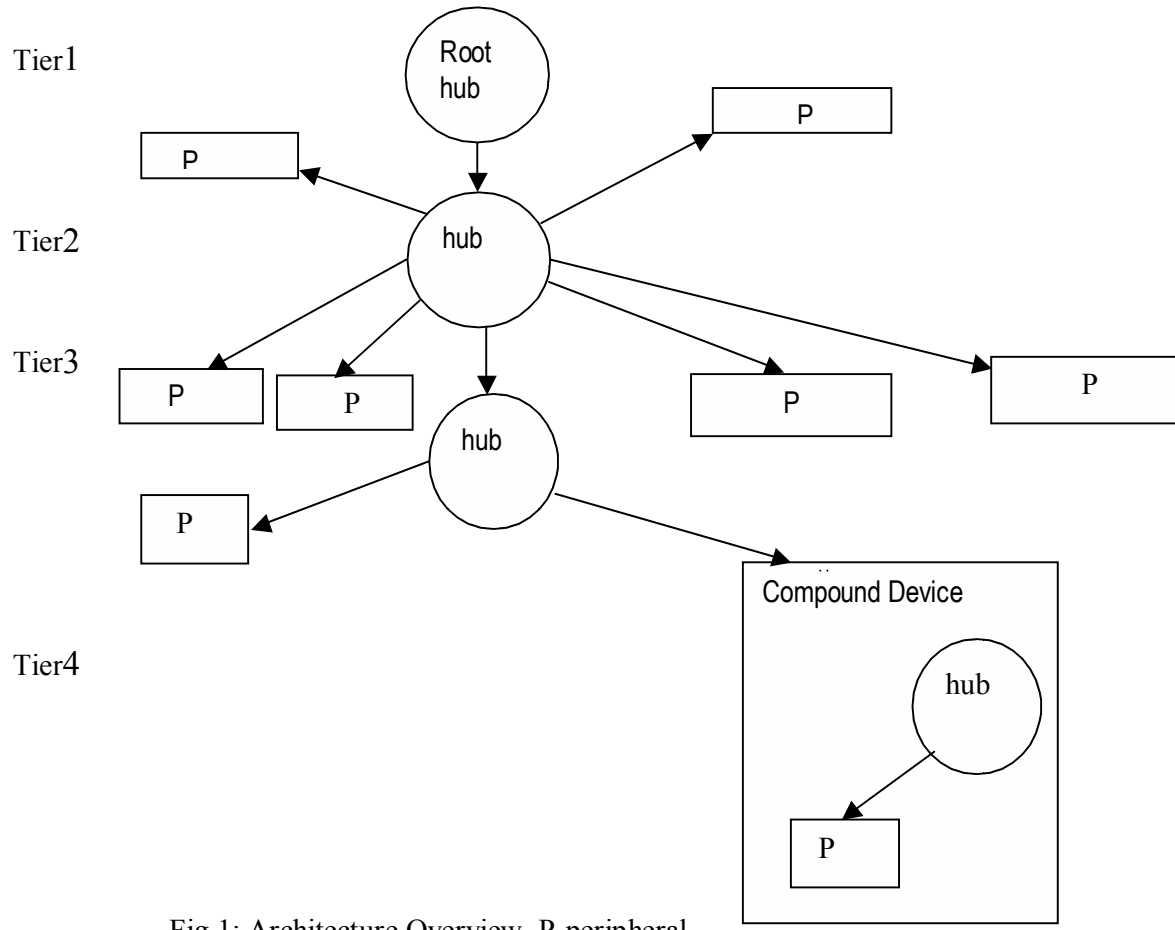


Fig 1: Architecture Overview, P-peripheral

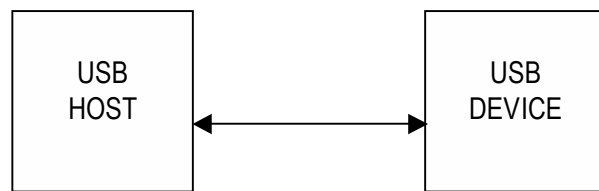


Fig 2: Host-Device view

But a very simple connection between host and peripheral appears for a user as shown in Fig 2. But, on the backside a number of interactions take place between different layers namely functional layer, device layer and USB interface layer. Actual communication flow and logical flow has shown in Fig 3. But a host communicates with each logical device (L.D) as if each logical device is connected to root port as shown in Fig 4. Even though physical and logical topology of USB reflects shared nature of bus, client software (C.S) manipulating a USB function interface is presented with the view that it deals only with the interface of interest. The figures described so far shows USB is host centric that is intervention of host PC is must. USB-IF released a new set of specifications (supplementary specifications) of USB 2.0. called On -The -Go (OTG) which provides point to point connectivity on September 2001 and its revision on December 2001. These specifies USB controller to

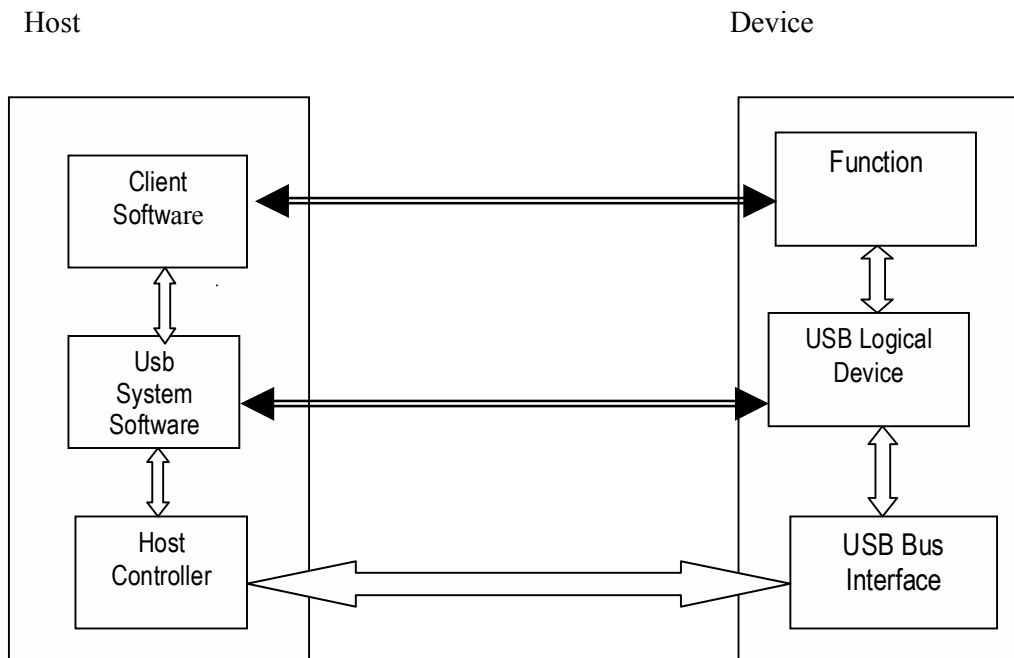
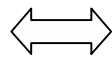
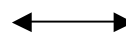


Fig 3: USB Data flow model



Data flow



Logical flow

play a dual role that is host for one device and peripheral for another one either one at a time or simultaneously. These controllers must support full speed transfer while high speed is optional. These controllers are useful in mobiles and embedded system

Inside USB controller

Controller chips vary in how much support they require for USB communications. Some controllers have a general purpose CPU on chip which controls the chip's actions by executing instructions stored in the firmware of the chip. The CPU may be microcontroller such as 8051 or it may be designed specifically for use in USB applications. Other controller chips interface to an external CPU. These chips support a command set for USB related communications. Therefore these chips provide a way to add USB capabilities to any microcontroller. The chips use external data buses (either serial or parallel) to connect to the CPU.

The main elements of USB controller are:

1) Program memory

It is the code CPU executes. The code configures the chip, access the USB port and also I/O pins. The memory may be ROM, RAM, EPROM etc. In case of RAM, the program must be loaded on power-up. For example Cypress Semiconductor's EZ-USB use RAM for program storage which gets loaded from PC on power-up or attachment to bus

2) *Data memory*

Contents of data memory include data received from USB devices, data to be sent to USB devices, values to be used in calculations like error checking, or anything which needs the chip to keep track of.

3) *Registers*

Status and control registers hold information about what end points are enabled, number of bytes received/transmitted, suspend status, error checking information.

4) *Serial interface engine (SIE)*

The circuits which control the USB interface form the serial interface engine. Some chips support all of USB transfer types while others support only a few transfer types.

5) *Buffers*

Transmit and receive buffers store USB data. Some chips use registers while others reserve a portion of data memory for the buffer.

Various types of USB controllers available from different manufactures are given below

- 1) Chips dedicated for USB applications –CY7C63xx (Cypress series)
- 2) Chips based on popular families- Intel 8x931 (8051 based), Intel 8x930 (80251 based), Cypress EZ-USB (8051 based)
- 3) Chips which interface to external CPU- National Semiconductor USBN9602, NET2888, Philips PD1USBD11/12

Advantages

User advantages

1) *Ease of use*

- One interface versatile enough to be usable with many kinds of peripherals
- Automatic configuration as operating system detects and loads the appropriate driver
- USB peripherals don't need user settings such as port addresses and IRQ lines (A non USB peripheral requires dedicated port addresses and often an IRQ line)
- Easy to connect as simple USB cable connectors namely A connectors and B connectors doesn't allow user to plug wrong
- Hot pluggable as user can connect/disconnect a peripheral without damaging PC or peripheral
- USB interface supplies power through one of its four pins to peripherals. So peripheral doesn't require a power supply of its own. A device which requiring less than 500mA can draw all its power from bus.

2) *Speed*

A wide variety of speeds suitable for almost all peripherals.

3) *Reliability*

The hardware specifications for USB drivers, receivers and cables eliminate most noise. USB protocol enables detecting of errors and notifies the sender so that it can retransmit.

4) *Low cost*

Though USB is more complex, a device with USB interface is likely to cost less. Its cables and components are inexpensive. Low speed devices have less stringent hardware requirements that may further reduce the cost.

5) *Low power consumption*

Power saving circuits and code automatically power down USB peripherals when there is no bus activity, yet keep them ready to respond when needed

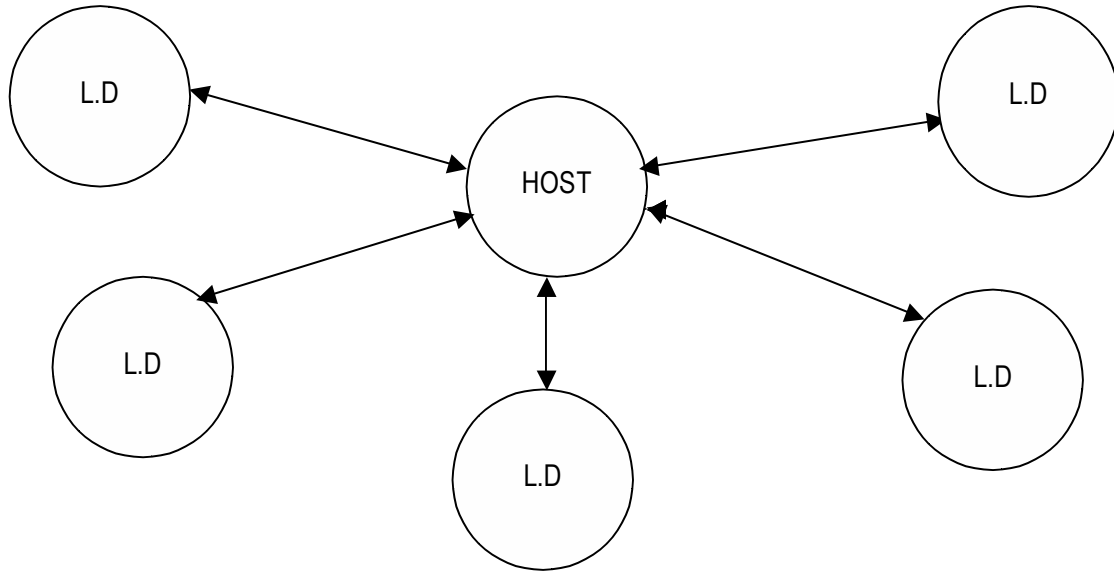


Fig 4: Host view of communication (L.D-logical device)

Developer advantages

The advantages of user make developer eager to develop USB peripherals as user buys them. In addition developer has the following :

1) *Easy to develop*

USB's defined cable standards and automatic error checking mean developers don't have to worry about specifying cable characteristics or providing error checking in software.

2) *Flexibility*

The flexibility built into USB protocol, the support in controller chips and the operating system, and the interface isn't controlled by single vendor. At a higher level, O.S. support may also mean the inclusion of software device drivers for some peripherals.

3) *Minimal fees*

USB specifications and related documents are available free online. Anyone can develop USB software without paying licensing fees. Of course to sell a device with USB interface, developer must sign an Adopters Agreement [1] and pay a fee for vendor ID.

Limitations

1) *Lack of support for legacy hardware*

A peripheral with non-USB interface support can be connected to USB port by using converters [2]. On the other hand, a peripheral with USB interface cannot be connected to older computers with no USB support. This can be possible only if host controller is plugged into PCI slot and operating system which supports USB is installed, both of which are expensive.

2) *Distance limits*

A cable segment can be as long as 5 m. It can be increased to 30 m by five hubs. Alternative solution is, USB interface on PC is converted to RS-485 interface using converters [2] for long distance cabling.

Data transfer types

The USB is designed to handle many types of peripherals with varying requirements for transfer rate, response time, and error correcting. USB supports four types of data transfer. A peripheral depending on its need can support any of the transfers. Every peripheral must also support Control Transfer along with its own data transfer support because enumeration of devices takes place through Control transfer type.

Types of Data transfer:-

1) *Control Transfer*

This type of data exchange is used only for enumeration and not for data transfer as it involves high protocol overhead. 10% of bandwidth is reserved for it.

2) *Interrupt Transfer*

This type of transfer is used mainly for human interface devices (HID) like mouse, keyboard etc. Latency is guaranteed that is after an interrupt is given host controller accepts the request within the specified time.

3) *Bulk transfer*

This type of transfer used where large amount of data transfer involves. This type of transfer is not used for time critical data. Guarantees data transfer. If busy is busy, it may take long time. If bus is idle, highest data transfer takes place. Suitable for printers, scanners etc.

4) *Isochronous transfer*

This type of data transfer is used for time critical data. Except for isochronous, all other data transfers has error checking code and retransmission facility. So isochronous devices must withstand wrong data transfers. This type data transfer devices are always provided enough bandwidth if host controller accepts configuration during enumeration. If busy is too busy, it ignores the device.

USB host controller executes data transfers in frames. A frame is 1 ms for low/full speed devices and 125 μ s for high speed devices. Each transfer is given a part of frame. Each transfer consists of one or more transactions. As all transfers share one data path, each transaction must include the address of the transaction's source or destination. A typical frame starts with start of frame (SOF), then few transfers depending on data packets width and ends with end of frame (EOF). Packet ID (PID) contains source/destination addresses, endpoint numbers and direction of data flow. A typical transfer is shown in Fig 5.

Enumeration

Enumeration is the initial exchange of information between host and device until the device driver starts communicating with the device.

Steps in enumeration

- 1) User plugs a device into a USB port. The port may be root hub at the host or on a hub.
- 2) The hub detects the device by monitoring the voltage on the signal lines of port. When a device plugs, the device pull-up brings that line high. It informs the host through interrupt pipe.
- 3) Host learns of new device and sends `Get_Port_Status` to find out more and Host-Controller sends `Set_Port_Feature` request to reset the port.
- 4) Hub establishes signal path between the device and the bus in the default state over the default pipe [3] at endpoint zero.
- 5) The hub detects the device speed by examining voltages on two signal lines and informs host. The host sends `Get_descriptor` request to learn maximum packet size of default pipe to device address zero [3], endpoint zero

- 6) Host reads first eight bytes of descriptor and host controller assigns a unique address by sending a Set_address request. The device reads request, stores the new address and now in address state.
- 7) The host learns about device abilities like the number of configurations the device supports and other basic information by sending Get_descriptor.
- 8) The host assigns and loads a device driver after it receives descriptors and finding the best match in device drivers.
- 9) The host device driver selects a configuration by sending a Set_configuration request. The device reads the request and sets its configuration to match and now device in configured state and device interface is enabled.

USB products [5]

1) Hubs

These are used to expand the peripherals connected. Hubs can be connected either to root hub or to another hub. The hub can provide either four or more ports depending on the manufacturer.

Model UH-91247 adds four more ports and its specifications are

- 4 downstream ports, USB type A receptacles
- 1 upstream port, type B receptacle
- 5 power/status indicator LEDs
- bus powered from USB host controller, or external power supply
- 6 ft USB cable included

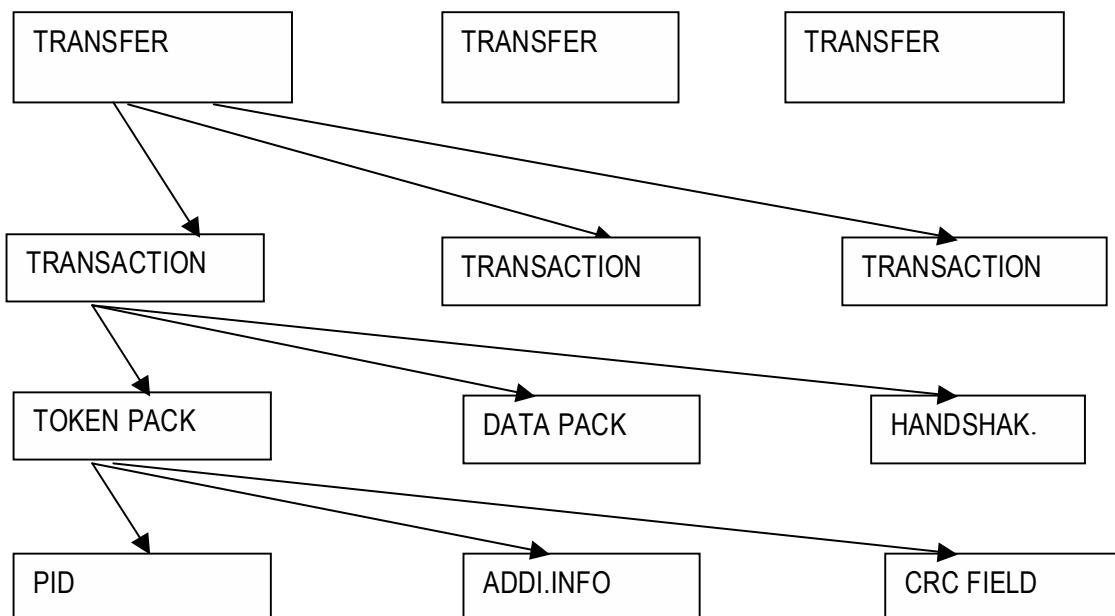


Fig 5 A typical transfer in a frame: CRC-cyclic redundancy check, PID-packet ID, ADDI.INFO-additional information; HANDSHAK-handshake packet

2) Converters

These are used to convert older interfaces into USB interfaces so that the peripheral can communicate with a USB port on a computer. A few models are shown

- USTOL4 optically isolated USB to RS-422/485.....\$139.95
- USTL4 non-isolated USB to RS-422/485.....\$119.95
- USO9ML2 optically isolated USB to RS-232.....\$139.95

3) Cables

These form the physical link between the host and the peripheral. USB supports a distance of 5m. If the cable characteristics does not meet strict specifications of USB errors occur during transmission. So host controller need to retransmit the data which is waste of time. Active cables solve this problem by buffering signals going in and out and thereby reducing errors.

- USB AEC16 USB active extender cable\$24.95 (active cable)
- USB AMBM-2M 2 meter type A male to type B male....\$8.95 (passive cable)
- USB AMBM-5M 5 meter type A male to type B male....\$9.95 (passive cable)

Conclusion

Presently, all the new PC's are coming with two USB ports. We can expand the number of peripherals by using hubs available in market [2]. We can upgrade older PC's by connecting Host-Controller in the expansion slot of motherboard provided operating system in PC has support for USB. Windows98 and its successors and many operating systems support it. In future, PC's will come with an embedded hub with varying number of ports with a minimum of four. As far as cost is concerned, it increases only slightly. IEEE 1394 parallel interface also known as peer-to-peer model is USB competitor. It has the capability of peripheral to peripheral communication. A single peripheral data can be directed to multiple receivers. IEEE 1394b has a speed of 3.2 Gbps almost six times of USB2.0. It has its use in very high bandwidth applications like video conferences. Its electronic circuitry is very complex and hence very expensive. Both the interfaces are likely to exist in tomorrow's PC's as USB is ideal for low/full speed and few high speed devices while IEEE1394 for large number of high speed devices.

References

1. <http://www.usb.org>, site sponsored by USB Implementers Forum, Inc, creators of USB technology, accessed on March 7, 2002
2. http://www.quatech.com/comm_main.htm#USB, accessed on Oct 1, 2002
3. Axelson, J., "*USB Complete*", Penram International Publishing (India), Mumbai 1999, vol. 1, ch.1, pp. 4, ch. 5 pp. 88, ch. 7 pp. 122-126.
4. <http://www.bb-elec.com/products.asp?Dept-id=145>, accessed on Oct 20, 2002.