19

The Electrical Interface

All of the protocols and program code in the world are no use if the signals don't make it down the cable in good shape. The electrical interface plays an important part in making USB a reliable way to transfer information.

From a practical point of view, if you're using compliant cables and components, you don't need to know much about the electrical interface. But if you're designing USB transceivers or cables, printed-circuit boards with USB interfaces, or a protocol analyzer that must unobtrusively monitor the bus, you do need to understand the electrical interface and how it affects the components in your project.

This chapter presents the essentials about the electrical interface of the USB's drivers and receivers and details about the cables that carry the signals.

Transceivers and Signals

The electrical properties of the signals on a USB cable vary depending on the speed of the cable segment. Low-, full-, and high-speed signaling each have a different edge rate, which is a measure of the rise and fall times of the voltages on the lines and thus the amount of time required for an output to switch. The transceivers and supporting circuits that produce and detect the bus signals also vary depending on speed.

At any speed, the components that connect to a USB cable must be able to withstand the shorting of any line to any other line or the cable shield without component damage.

Cable Segments

A cable segment is a single physical cable that connects a device (which may be a hub) to an upstream hub (which may be the root hub at the host). The speed, edge rate, and polarity of the data in a segment depend on whether the segment is low, full, or high speed. Figure 19-1 illustrates.

Low-speed segments exist only between low-speed devices and their hubs. A low-speed segment carries only low-speed data, using low-speed's edge rate and inverted polarity compared to full speed.

A full-speed segment exists when the segment's downstream device is operating at full speed. The upstream device may be a 1.x or 2.0 hub (including the root hub). When the downstream device is a hub, the segment may also carry data to and from low-speed devices that are downstream from that hub. In this situation, the low-speed data on the full-speed segment uses low-speed's bit rate but full speed's polarity and edge rate. The hub that connects to the low-speed device converts between low and full speed's polarity and edge rates. Full-speed segments never carry data at high speed. If a high-speed-capable device connects to a 1.x hub, communications are at full speed. High-speed devices must at least respond to enumeration requests at full speed.

High-speed segments exist only where the host is USB 2.0, all upstream device(s) are 2.0 hubs, and the downstream device is high speed. When the

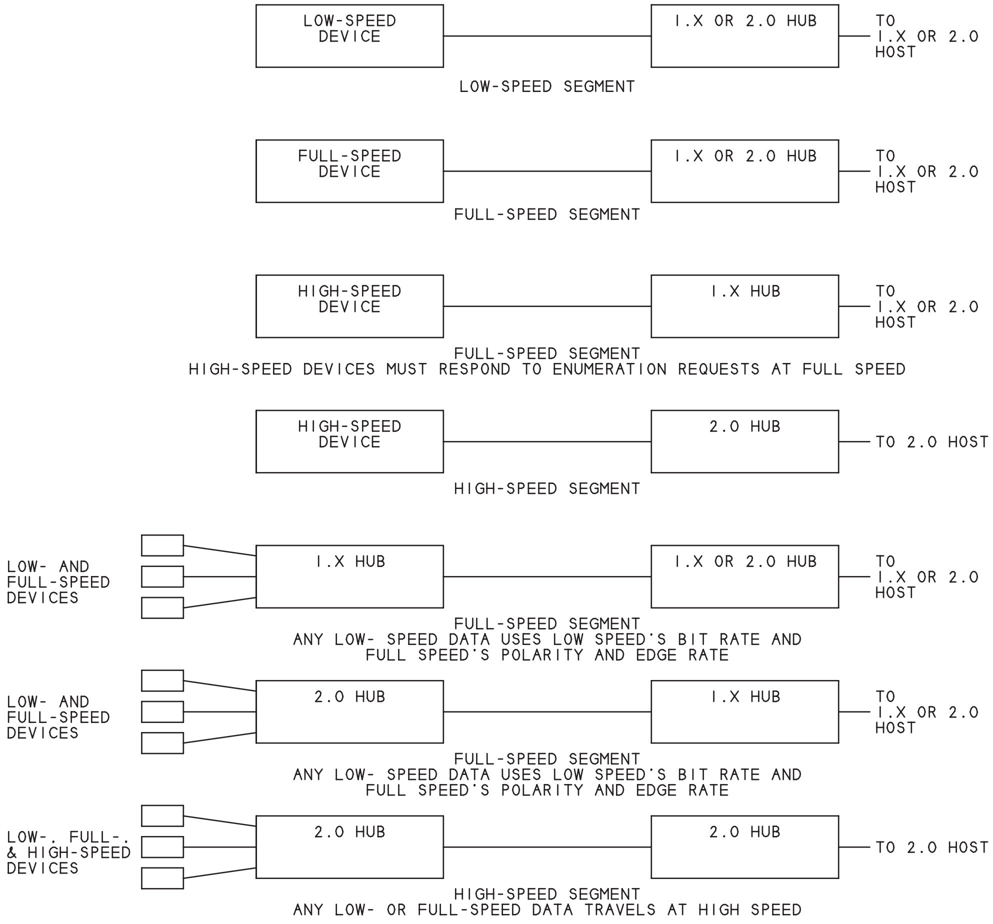

Figure 19-1: The speed of data in a segment depends on the capabilities of the device and its upstream hub.

downstream device is a hub, the segment may also carry data to and from low- and full-speed devices that are downstream from that hub. All data in a high-speed segment travels at high speed, and the transaction translator in a downstream hub converts between low or full speed and high speed as needed.

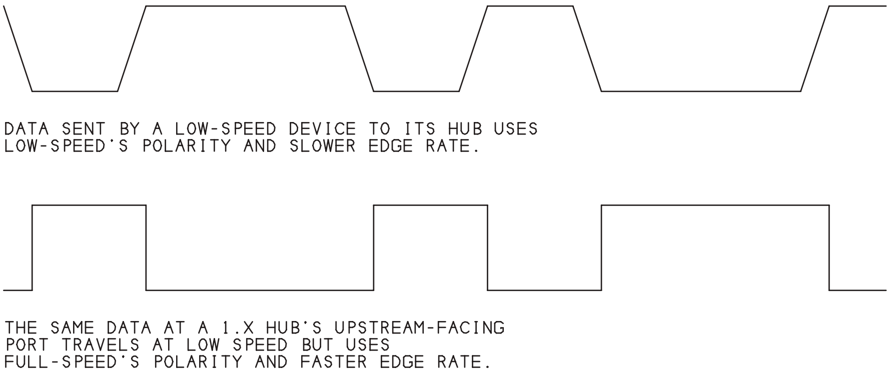

Figure 19-2: A 1.x hub converts between low- and full-speed's polarities and edge rates. (Not drawn to scale)

On attachment, all devices must communicate at low or full speed. When possible, a high-speed-capable device transitions from full to high speed shortly after the device is attached, during the high-speed handshake.

Low- and Full-speed Transceivers

The transceiver for low and full speeds has a simpler design compared to the transceiver for high speed.

Low- and Full-speed Differences

Low-speed data differs electrically from full speed in three ways. The bit rate is slower, at 1.5 Megabits/sec. compared to 12 Megabits/sec. for full speed. Low speed traffic's polarity is inverted compared to full speed. And low speed has a slower edge rate compared to full speed. Figure 19-2 illustrates. The slower edge rate reduces reflected voltages on the line and makes it possible to use cables that have less shielding and are thus cheaper to make and physically more flexible.

The transceiver's hardware doesn't care about the signal polarity. The transceiver just retransmits whatever logic levels are at its inputs. A driver that supports both speeds, such as a driver for a hub's downstream port, must be able to switch between the two edge rates.

The Circuits

Figure 19-3 shows port circuits and cable segments for low- and full-speed communications. Each transceiver contains a differential driver and receiver for sending and receiving data on the bus's twisted pair.

When transmitting data, the driver has two outputs that are 180 degrees out of phase: when one output is high, the other is low. A single driver can support both low and full speeds with a control input to select the full-speed or low-speed edge rate.

The differential receiver detects the voltage difference between the lines. A differential receiver has two inputs and defines logic levels in terms of the voltage difference between the inputs. Some differential interfaces, such as RS-485, define logic levels strictly as the difference between voltages on the two signal lines, with no reference to ground (though the interface requires a common ground connection). USB differs because it specifies absolute voltages in addition to a required voltage difference at the receivers. The differential receiver's output is a logic-high or logic-low voltage referenced to ground.

Each port also has two single-ended receivers that detect the voltages on D+ and D- with reference to signal ground. The logic states of the receivers' outputs indicate whether the bus is low or full speed or whether the bus is in the Single-Ended-Zero state.

The drivers' output impedances plus a 36-ohm series resistor at each driver's output act as source terminations that reduce reflected voltages when the outputs switch. The series resistors may be on-chip or external to the chip.

Pull-up and Pull-down Values

The pull-up resistor on D+ or D- at a device's upstream-facing port enables the hub to detect the device's speed. The hub's downstream-facing port has pull-down resistors on D+ and D-.

On devices with detachable cables, the pull-up resistors must connect to a voltage source of 3.0–3.6V. Devices with captive cables can instead use an

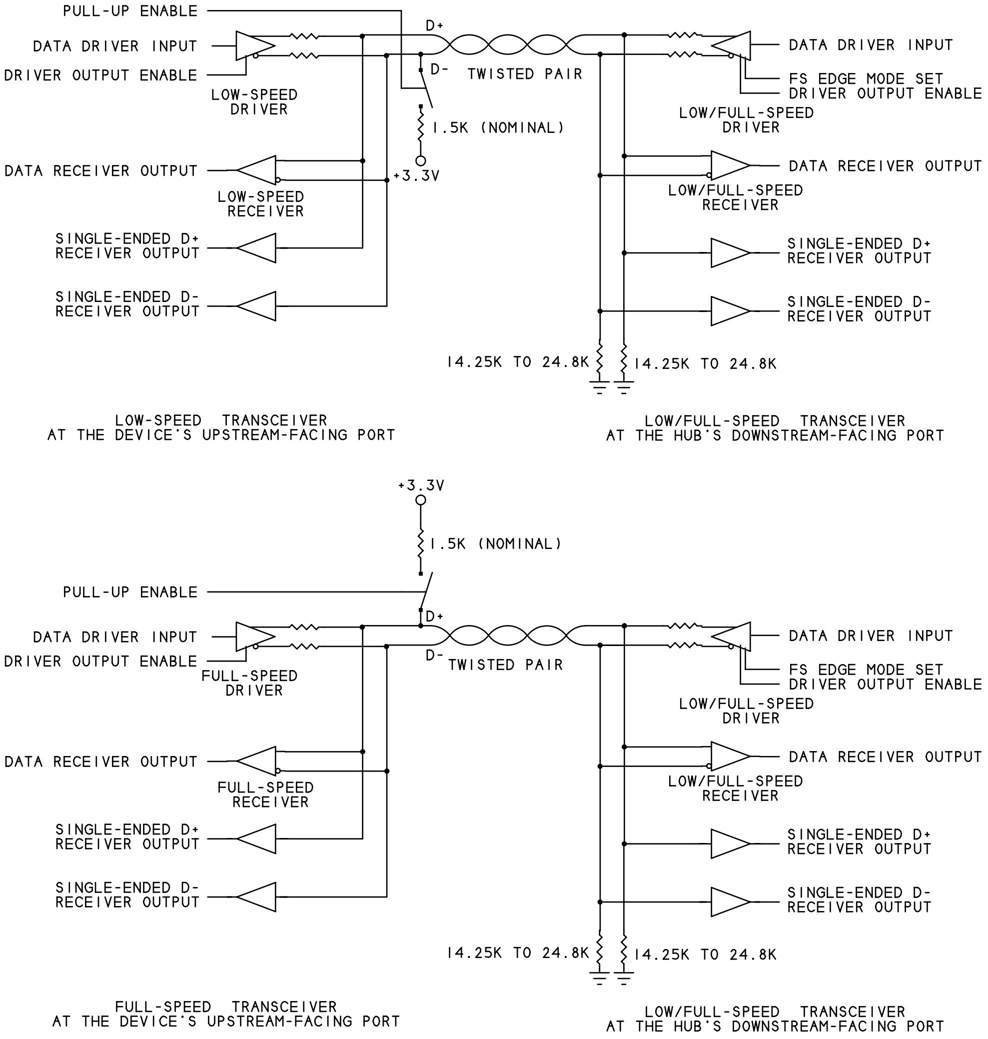

Figure 19-3: The downstream-facing ports on a 1.x hub must support both low and full speeds (except for ports with embedded or permanently attached devices). A device’s upstream-facing port typically supports just one speed.

alternative means of termination, including connecting directly to VBUS. In selecting an alternatative means of termination, the designer is responsible for ensuring that all of the bus's signal levels meet the USB specification's requirements.

An Engineering Change Notice titled *Pull-up/pull-down resistors* revises the USB 2.0 specification by loosening the tolerances for pull-up and pull-down resistors that connect to a voltage source of 3.0–3.6V. The original values were 1.5 kilohms ±5% for the pull ups and 15 kilohms ±5% for the pull downs. The tolerances were loosened to make it easier to include the resistors on chip without requiring laser trimming of the values. Using the looser tolerances increases complexity slightly at upstream-facing ports because the device must switch between two pull-up values depending on whether the bus is idle or active. But overall, the result can be reduced cost to device manufacturers.

Table 19-1 shows the new values. Devices that use the old tolerances remain compliant, and devices that use the old tolerances can communicate with devices that use the new tolerances. To use the wider tolerances, a device must use one pull-up value when the bus is idle and switch to a higher value when the upstream device begins to transmit. The upper limit on the pull up for the idle bus ensures that the idle voltage is at least the required minimum of 2.7V. For the active bus, the lower limit is the same as the original lower limit and the upper limit ensures that the data line remains in a high state if the receiver interprets noise as a Start-of-Packet signal.

Using the new limits, the resistors can have tolerances as high as 27%. Examples of compliant values are 19 kilohms ±25% for the pull downs and 1200 and 2400 ohms ± 25% for the pull ups. A device can implement its pull up using two resistors in series, switching the second resistor into the circuit when the upstream device begins to transmit. A device must switch to the higher resistance within 0.5 bit time of detecting a J-to-K transition on the bus. To determine when to switch to the lower resistance, a device may use either or both of the following methods: on detecting a Single-ended Zero for more than 0.5 bit time or on detecting that the bus has been in the J state for more than 7 bit times. The ECN details a few hardware implications for designers of chips that use the wider tolerances.

Table 19-1: Values for the pull-up and pull-down resistors at the device and hub. The pull-up values assume that the pull up connects to a voltage source of 3–3.6V, as required for devices with detachable cables.

Resistor	Bus State	Minimum (ohms)	Maximum (ohms)	Acceptable Value with 25% Tolerance
pull down	All	14,250	24,800	19k
pull up	Idle	900	1575	1.2k
	Active	1425	3090	2.4k
	Single-Ended Zero	900	<3090	2.4k

High-speed Transceivers

A high-speed device must support control requests at full speed, so the device must contain transceivers to support both full and high speeds and the logic to switch between them. A high-speed-capable device's upstream transceivers aren't allowed to support low speed. In an external 2.0 hub, the downstream transceivers at ports with user-accessible connectors must support all three speeds.

Why 480 Megabits per Second?

High speed's rate of 480 Megabits/sec. was chosen for several reasons. The frequency is slow enough to allow using the same cables and connectors as full speed. Components can use CMOS processes and don't require the advanced compensation used in high-speed digital signal processors. Tests of high-speed drivers showed 20 to 30 percent jitter at 480 Megabits/sec. Because receivers can be designed to tolerate 40 percent jitter, this bit rate allows a good margin of error. And 480 is an even multiple of 12, so a single crystal can support both full and high speed.

The use of separate drivers for high speed makes it easy to add high speed to the existing interface. Current-mode drivers were chosen because they're fast.

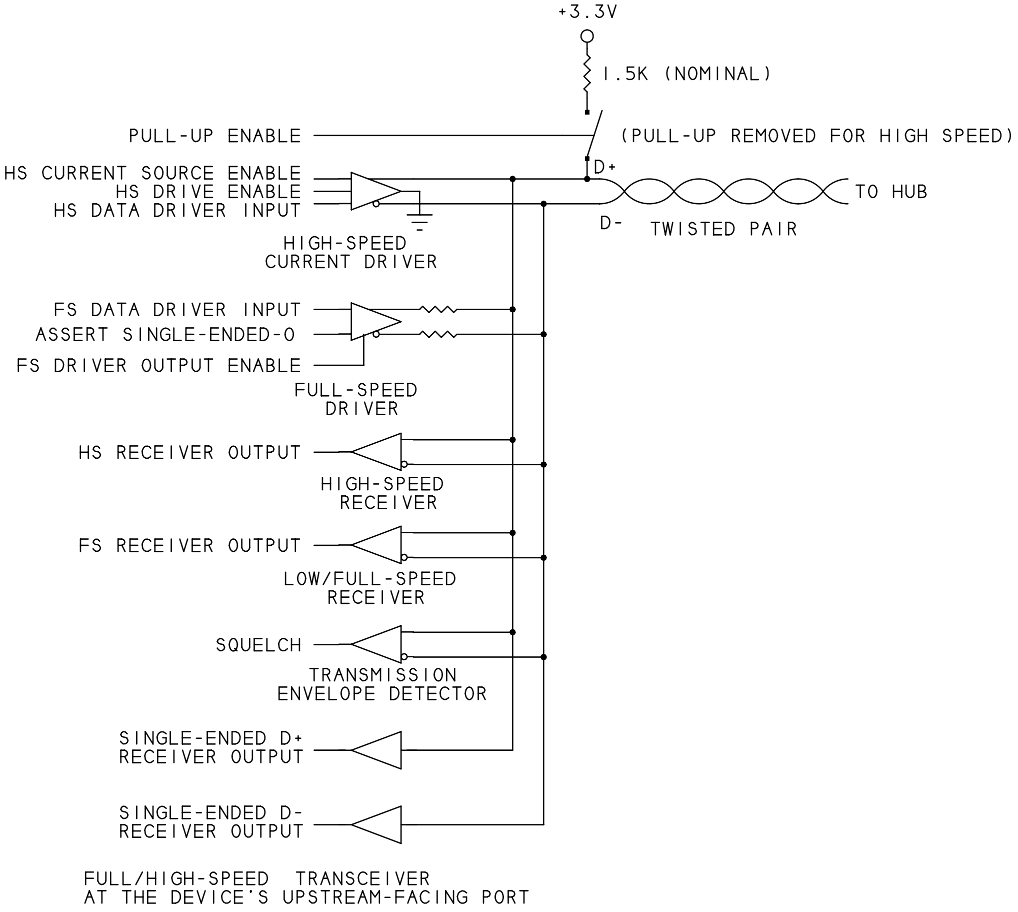

Figure 19-4: The upstream-facing port on a high-speed device must also support full-speed communications.

The Circuits

Figure 19-4 shows upstream-facing transceiver circuits in a high-speed-capable device, and Figure 19-5 shows downstream-facing transceiver circuits in a 2.0 hub.

High speed requires its own drivers, so a high-speed device must contain two sets of drivers. For receiving, a transceiver may use a single receiver to

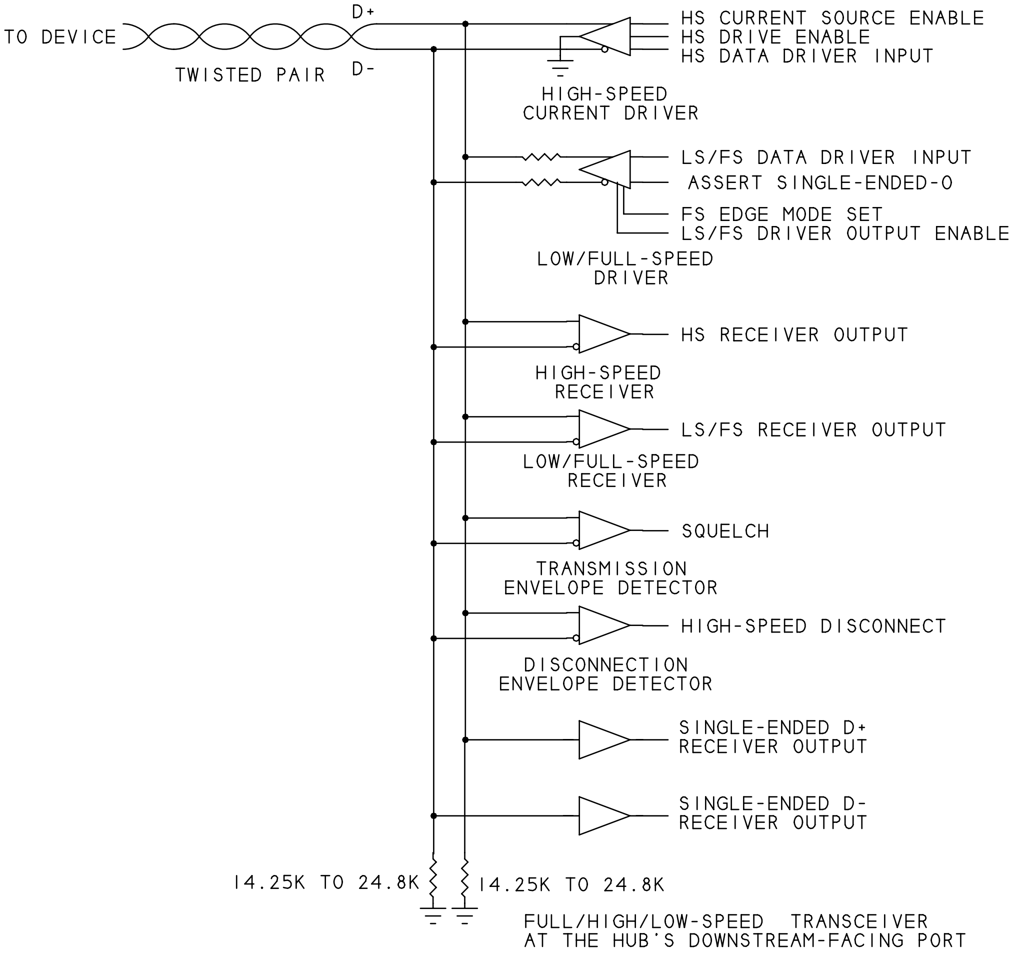

Figure 19-5: The downstream-facing ports on external 2.0 hubs must support all three speeds (except for ports with embedded or permanently attached devices).

handle all supported speeds or separate receivers for low/full speed and high speed.

When a high-speed driver sends data, a current source drives one line with the other line at ground. The current source may be active all the time or only when transmitting. A current source that is active all the time is easier to design but consumes more power. The USB specification requires devices to meet the signal-amplitude and timing requirements beginning with the first symbol in a packet. This requirement complicates the design of a current source that is active only when transmitting. If the driver instead keeps its current source active all the time, the driver can direct the current to ground when not transmitting on the bus.

In a high-speed-capable transceiver, the output impedance of the full-speed drivers has tighter tolerance compared to full-speed-only drivers (45 ohms ±10%, compared to 36 ohms ±22%). The change is required because the high-speed bus uses the full-speed drivers as electrical terminations on the cable. Full-speed drivers that aren't part of a high-speed transceiver don't require a change in output impedance.

When the high-speed drivers are active, the full-speed drivers bring both data lines low (the Single-ended-Zero state). Each driver and its series resistor then function as a 45-ohm termination to ground. Because there is a driver at each end of the cable segment, there is a termination at both the source and the load. This double termination quiets the line more effectively than the source-only series terminations in full-speed segments. Using the full-speed drivers as terminations means no extra components are required.

The USB specification provides eye-pattern templates that show the required high-speed transmitter outputs and receiver sensitivity. High-speed receivers must also meet new specifications that require the use of a differential time-domain reflectometer (TDR) to measure impedance characteristics.

All high-speed receivers must include a differential envelope detector to detect the Squelch (invalid signal) state, indicated by a differential bus voltage of 100 millivolts or less. The downstream ports on all 2.0 hubs must also include a high-speed-disconnect detector that detects when a device has been removed from the bus.

Other new responsibilities for high-speed-capable devices include managing the switch from full to high speed and handling new protocols for entering and exiting the Suspend and Reset states.

Switching Speeds

In a low- or full-speed device, a pull-up resistor on one of the signal lines indicates device speed. When a low- or full-speed device is attached or removed from the bus, the voltage change due to the pull up informs the hub of the change. High-speed-capable devices always attach at full speed, so hubs detect attachment of high-speed-capable devices in the same way.

As Chapter 18 explained, the switch to high speed occurs after the device has been detected, during the Reset sent by the hub. A high-speed-capable device must support the high-speed handshake that informs the hub that the device is capable of high speed. When switching to high speed, the device removes its pull up from the bus.

Detecting Removal of a High-speed Device

A 2.0 hub must also detect the removal of a high-speed device. Because the device has no pull up at high speed, the hub has to use a different method to detect the removal. When a device is removed from the bus, the differential terminations are removed, and the removal causes the differential voltage at the hub's port to double. On detecting the doubled voltage, the hub knows the device has been removed.

The hub detects the voltage by measuring the differential bus voltage during the extended End of High-speed Packet (HSEOP) in each high-speed Start-of-Frame Packet (HSSOP). A differential voltage of at least 625 millivolts indicates a disconnect.

Suspending and Resuming at High Speed

As Chapter 16 explained, devices must enter the low-power Suspend state when the bus has been in the Idle state for at least 3 milliseconds and no more than 10 milliseconds. When the bus has been idle for 3 milliseconds, a high-speed device switches to full speed. The device then checks the state of the full-speed bus to determine whether the host is requesting a Suspend or Reset. If the bus state is Single-Ended Zero, the host is requesting a Reset, so the device prepares for the high-speed-detect handshake. If the bus state is Idle, the device enters the Suspend state. The device must return to high speed on exiting the Suspend state.

Signal Voltages

Chapter 18 introduced USB's bus states. The voltages that define the states vary depending on the speed of the cable segment. The differences in the specified voltages at the transmitter and receiver mean that a signal can have some noise or attenuation and the receiver will still see the correct logic level.

Low and Full Speeds

Table 19-2 shows the driver output voltages for low/full and high speeds. At low and full speeds, a Differential 1 exists at the driver when the D+ output is at least 2.8V and the D- output is no greater than 0.3V referenced to the driver's signal ground. A differential 0 exists at the driver when D- is at least 2.8V and D+ is no greater than 0.3V referenced to the driver's signal ground.

At a low- or full-speed receiver, a differential 1 exists when D+ is at least 2V referenced to the receiver's signal ground, and the difference between D+ and D- is greater than 200 millivolts. A differential 0 exists when D- is at least 2V referenced to the receiver's signal ground, and the difference between D- and D+ is greater than 200 millivolts. However, a receiver may optionally have less stringent definitions that require only a differential voltage greater than 200 millivolts, ignoring the requirement for one line to be at least 2V.

Table 19-2: High speed requires different drivers and has different output specifications, compared to low and full speed. The receiver specifications differ as well.

Parameter	**Low/Full Speed (V)**	**High Speed (V)**
Vout low minimum	0	-0.010
Vout low maximum	0.3	0.010
Vout high minimum	2.8	0.360V
Vout high maximum	3.6	0.440V
Vin low maximum	0.8	Limits are defined by the eye-pattern templates in the USB specification
Vin high minimum	2.0	

High Speed

At high speed, a differential 1 exists at the driver when the D+ output is at least 0.36V and the D- output is no greater than 0.01V referenced to the driver's signal ground. A differential 0 exists at the driver when D- is at least 0.36V and D+ is no greater than 0.01V referenced to the driver's signal ground.

At a high-speed receiver, the input must meet the requirements shown in the eye-pattern templates in the USB specification. The eye patterns specify maximum and minimum voltages, rise and fall times, maximum jitter in a transmitted signal, and the maximum jitter a receiver must tolerate. The USB specification has details about how to make the measurements.

Cables

The USB 2.0 specification includes detailed requirements for cables. The requirements help to ensure that any compliant cable will be able to carry the bus's digital signals without errors due to noise in the cable without large amounts of noise radiating from the cable.

Conductors

USB cables have four conductors: VBUS, GND, D+ and D-.

VBUS is the +5V supply.

GND is the ground reference for VBUS as well as for D+ and D-.
D+ and D- are the differential signal pair.

Chapter 16 described the voltage and current limits for VBUS.

Cables to be used in full- or high-speed segments have different requirements compared to cables for low-speed segments. Table 19-3 compares the two cable types. A low-speed segment is a cable segment between a low-speed device and its hub. Any additional upstream segments between hubs are considered to be full- or high-speed segments.

The USB 2.0 specification tightened the requirements for low-speed cables. A 1.1-compliant low-speed cable required no shielding at all. A 2.0-compliant low-speed cable must have the same inner shield and drain wire required for full speed. The USB specification also recommends, but doesn't require, a braided outer shield and a twisted pair for data, as on full- and high-speed cables.

Full- and high-speed segments can use the same cables. When the USB 2.0 specification was under development, an Engineering Change Notice to the 1.x specification added new requirements to ensure that full-speed cables would also work at high speed. The 2.0 specification also includes these requirements. The requirements describe what was typically found in compliant full-speed cables, so most providers with compliant cables had no changes to make to their products.

In a full/high-speed cable, the signal wires must have a differential characteristic impedance of 90 ohms. This value is a measure of the input impedance of an infinite, open line and determines the initial current on the lines when the outputs switch. The characteristic impedance for a low-speed cable isn't defined because the slower edge rates mean that the initial current doesn't affect the logic states seen by the receiver.

The USB specification lists requirements for the cable's conductors, shielding, and insulation. These are the major requirements for full/high-speed cables:

Data wires: twisted pair, #28 AWG.
Power and ground: non-twisted, #20 to #28 AWG.

Table 19-3: The requirements for cables and related components differ for full/high-speed cables and cables that attach to low-speed devices.

Specification	**Low Speed**	**Full/High Speed**
Maximum length (meters)	3	5
Inner shield and drain wire required?	yes (new in USB 2.0)	yes
Braided outer shield required?	no, but recommended	yes
Twisted pair required?	no, but recommended	yes
Common-mode impedance (ohms)	not specified	30 ±30%
Differential Characteristic impedance (ohms)	not specified	90
Cable skew (picoseconds)	< 100	
Wire gauge (AWG#)	20 –28	
DC resistance, plug shell to plug shell (ohms)	0.6	
Cable delay	18 nanosecs. (one way)	5.2 nanoseconds/meter
pull up location at the device	D-	D+
Detachable cable OK?	no	yes
Captive cable OK?	yes	

Drain wire: stranded, tinned copper wire, #28 AWG
Inner shield: aluminum metallized polyester
Outer shield: braided, tinned copper

The USB specification also lists requirements for the cable's durability and performance.

A low-speed device can use a full-speed cable if the cable meets all of the low-speed cable requirements. These include *not* using any standard USB connector type at the device end and a maximum length of 3 meters.

Connectors

The USB specifications define four plug types for USB cables. USB 2.0 defines the Series-A plug for the upstream end of the cable and the Series-B plug for the downstream end of the cable. Each plug type has a mating receptacle type. (Figure 19-6). Because the Series-B connectors were bulky for some devices, a new mini-B connector was defined in an Engineering Change Notice titled *Mini-B connector*. A mini-B receptacle is less than half the height of a Series-B receptacle. Any device can use a mini-B receptacle instead of a Series-B receptacle. The On-The-Go supplement adds a mini-A plug as an option for connecting to On-The-Go hosts. Figure 19-7 shows all four plug types. Chapter 20 has more about On-The-Go connectors.

All of the connectors are keyed so you can't insert a plug upsidedown. The signal connections are recessed slightly to ensure that the power lines con-

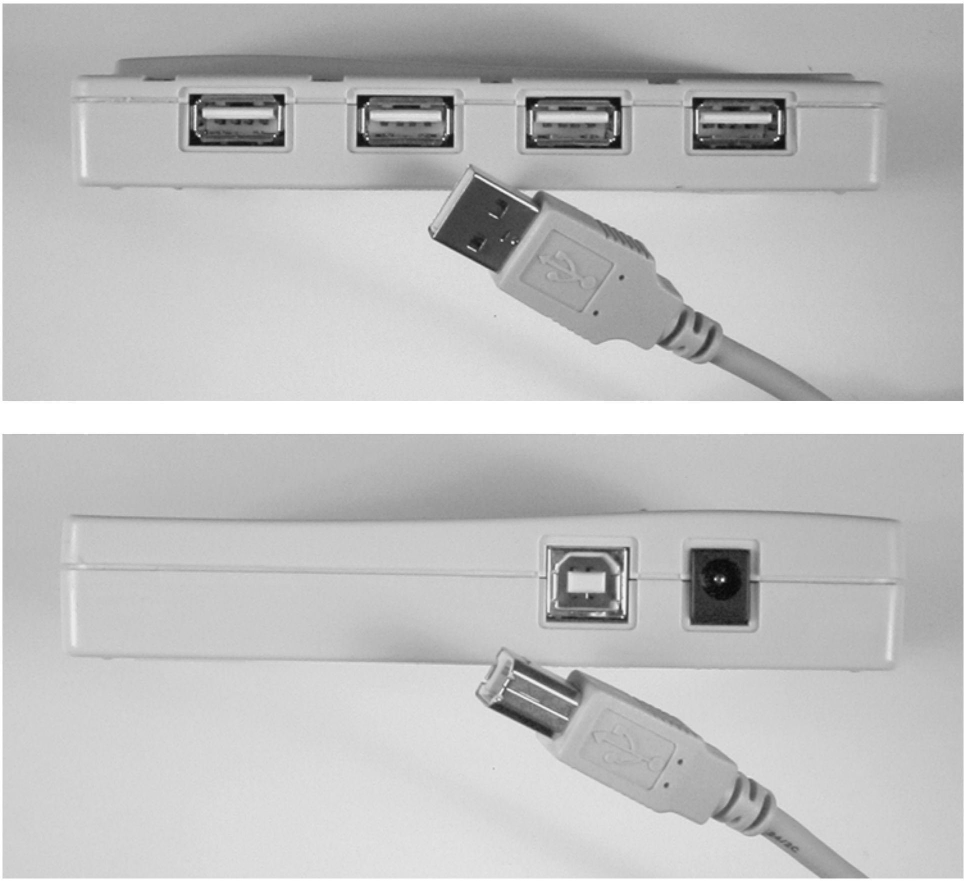

Figure 19-6: The Series-A plug (top) is on the upstream end of the cable and mates with a Series-A receptacle on a hub or the host. The Series-B plug (bottom) is on the downstream end of the cable and mates with a Series-B receptacle on the device.

nect first when a cable is attached. The receptacle should be mounted so the USB icon on the top of the plug is visible when a plug is attached.

The USB icon can identify a USB plug or receptacle (Figure 19-8). A "+" added to the icon indicates that a downstream-facing port supports high speed. Don't confuse the icon with the USB logo described in Chapter 17.

All of the connectors have connections for the bus's two signal wires, the VBUS supply, and ground. The mini-A and mini-B plugs have an additional ID pin. On-The-Go devices use the ID pin to identify a device's default

Figure 19-7: The USB specifications define four plug types. From left to right, they are Series A, Series B, mini-A, and mini-B.

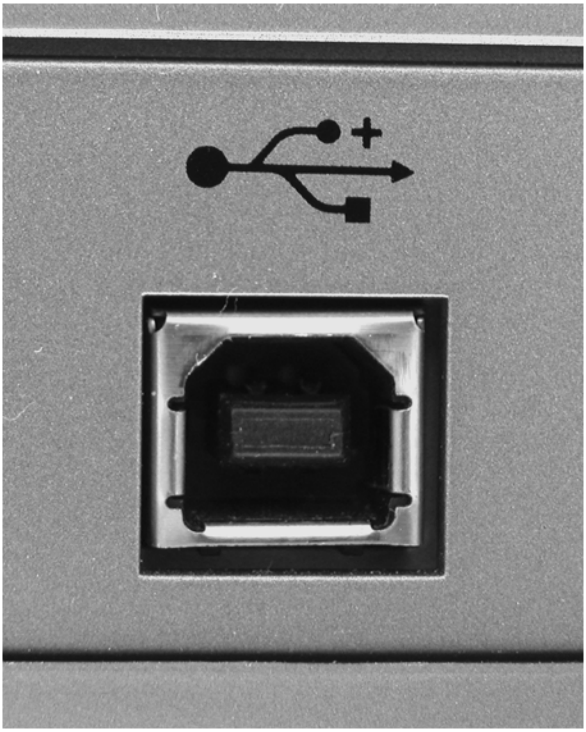

Figure 19-8: The USB icon identifies a USB plug or receptacle. A "+" indicates support for high speed.

mode (host or function). The USB 2.0 specification gives the following pin and color assignments for the cable and connectors:

Series A or Series B pin	**Mini-B pin**	**Conductor**	**Cable Wire**
1	1	VBUS (+5V)	red
2	2	D-	white
3	3	D+	green
4	5	GND	black
-	4	ID	not connected
shell		shield	drain wire

Detachable and Captive Cables

The USB specification defines cables as being either detachable or captive. From the names, you might think that a detachable cable is one you can remove, while a captive cable is permanently attached to its downstream device. But in fact, a captive cable can be removable as long as its downstream connector is *not* one of the standard USB connector types.

A detachable cable must be full/high speed, with a Series-A plug for the upstream connection and a Series-B or mini-B plug for the downstream connection. A captive cable may be low or full/high speed. The upstream end has a Series-A plug. For the downstream connection, the cable can be permanently attached or removable with a non-standard connector type. The non-standard connector doesn't have to be hot pluggable, but the Series-A connector must be hot pluggable. Requiring low-speed cables to be captive eliminates the possibility of trying to use a low-speed cable in a full- or high-speed segment.

Cable Length

Version 1.0 of the USB specification gave maximum lengths for cable segments. A full-speed segment could be up to 5 meters and a low-speed segment could be up to 3 meters. Version 1.1 dropped the length limits in favor of a discussion of characteristics that limit a cable's ability to meet the interface's timing and voltage requirements. On full- and high-speed cables, the limits are due to signal attenuation, cable propagation delay (the amount of time it takes for a signal to travel from driver to receiver), and the voltage drops on the VBUS and GND wires. On low-speed cables, the length is limited by the rise and fall times of the signals, the capacitive load presented by the segment, and the voltage drops on the VBUS and GND wires.

The original limits of 3 and 5 meters are still good guidelines. A 2.0-compliant 5-meter cable will work at full and high speeds. Compliant cables of these lengths are readily available. Chapter 16 explained how the length limits translate to a maximum distance of 30 meters between a host and its peripheral, assuming the use of five hubs and six 5-meter cable segments.

The USB specification prohibits extension cables, which would extend the length of a segment by adding a second cable in series. An extension cable for the upstream side of a cable would have a Series-A plug on one end and a Series-A receptacle on the other, while an extension cable for the downstream side would have a Series-B plug and receptacle.

Prohibiting extension cables eliminates the temptation to stretch a segment beyond the interface's electrical limits. Extension cables are available, but just because you can buy one doesn't mean that it's a good idea or that the cable will work. Instead, buy a single cable of the length you need and add hubs as needed.

An exception is an active extension cable that contains a hub, a downstream port, and a cable. This type of cable works fine because it contains the required hub. Depending on the attached devices, the hub may need its own power supply. Chapter 20 discusses two cable adapters that are approved for use only with On-The-Go devices.

An option for longer distances is to use a standard USB cable that connects to a device that translates between USB and Ethernet, RS-485, or another interface designed for use over long distances. The remote device would then need to support the long-distance interface, rather than USB.

Another option enables you to place a USB device anywhere in a local Ethernet network. Two products that use this approach are the AnywhereUSB hub from Inside Out Networks, Inc. and the USB Server from Keyspan. The hub/server contains one or more host controllers that communicate with the host PC over an Ethernet connection using the Internet Protocol (IP). The hub/server can attach to any Ethernet port in the PC's local network. The device drivers are on the PC. The PC can use the hub/server to access many devices that use bulk and interrupt transfers, with some increased latency due to the additional protocol layer.

Ensuring Signal Quality

The USB specifications for drivers, receivers, and cable design ensure that virtually all data transfers occur without errors. Requirements that help to ensure signal quality include the use of balanced lines and shielded cables, twisted pairs required for full/high-speed cables, and slower edge rates required for low-speed drivers.

Sources of Noise

Noise can enter a wire in many ways, including by conductive, common-impedance, magnetic, capacitive, and electromagnetic coupling. If a noise voltage is large enough and is present when the receiver is attempting to detect a transmitted bit, the noise can cause the receiver to misread the received logic level. Very large noise voltages can damage components.

Conductive and common-impedance coupling require ohmic contact between the signal wire and the wire that is the source of the noise. Conductive coupling occurs when a wire brings noise from another source into a circuit. For example, a noisy power-supply line can carry noise into the circuit the supply powers. Common-impedance coupling occurs when two circuits share a wire, such as a ground return.

The other types of noise coupling result from interactions between the electric and magnetic fields of the wires themselves and signals that couple into the wires from outside sources, including other wires in the interface. Capacitive and inductive coupling can cause crosstalk, where signals on one wire enter another wire. Capacitive coupling, also called electric coupling, occurs when two wires carry charges at different potentials, resulting in an electric field between the wires. The strength of the field and the resulting capacitive coupling varies with the distance between the wires. Inductive, or magnetic, coupling occurs because current in a wire causes the wire to emanate a magnetic field. When the magnetic fields of two wires overlap, the energy in each wire's field induces a current in the other wire. When wires are greater then 1/6 wavelength apart, the captive and inductive coupling is considered together as electromagnetic coupling. An example of electromagnetic coupling is when a wire acts as a receiving antenna for radio waves.

Balanced Lines

One way that USB eliminates noise is with the balanced lines that carry the bus's differential signals. Balanced lines are electrically quiet. Noise that couples into the interface is likely to couple equally into both signal wires. At a differential receiver, which detects only the difference between the two wires' voltages, any noise that is common to both wires cancels out.

In contrast, in the unbalanced, single-ended lines used by RS-232 and other interfaces, the receiver detects the difference between a signal wire and a ground line shared by other circuits. The ground line is likely to be carrying noise from a number of sources, and the receiver sees this noise when it detects the difference between the signal voltage and ground.

Twisted Pairs

In a full/high-speed USB cable, the two signal wires must form a twisted pair. Twisted pairs are recommended, but not required, for low-speed cables. A twisted pair is two insulated conductors that spiral around each other with a twist every few inches (Figure 19-9). The twisting reduces noise in two ways: by reducing the amount of noise in the wires and by canceling whatever noise does enter the wires. Twisting is most effective at eliminating low-frequency, magnetically coupled signals such as 60-Hz power-line noise.

Twisting reduces noise by minimizing the area between the conductors. The magnetic field that emanates from a circuit is proportional to the area between the conductors. Twisting the conductors around each other reduces the total area between them. The tighter the twists, the smaller the area. Reducing the area shrinks the magnetic field emanating from the wires and thus reduces the amount of noise coupling into the field.

A twisted pair tends to cancel any noise that enters the wires because the conductors swap physical positions with each twist. Any noise that magnetically couples into the wires reverses polarity with each twist. The result is that the noise present in one twist is cancelled by a nearly equal, opposite

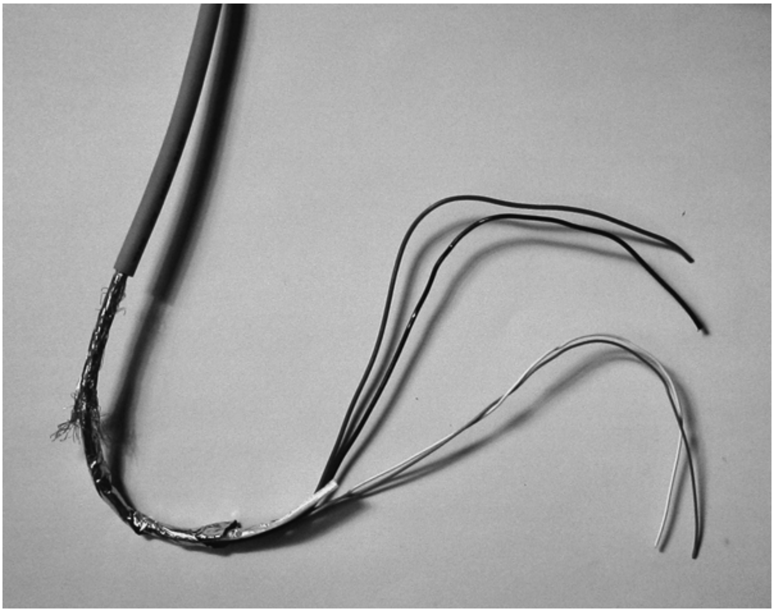

Figure 19-9: A full/high-speed USB cable contains a twisted pair for data, VBUS and GND wires, and aluminum metallized polyester and braided copper shields.

noise signal in the next twist. Of course, the twists aren't perfectly uniform, so the canceling isn't perfect, but noise is much reduced.

Shielding

Metal shielding prevents noise from entering or emanating from a cable. Shielding is most effective at blocking noise due to capacitive, electromagnetic, and high-frequency magnetic coupling. The USB 2.0 specification requires both low-speed and full/high-speed cables to be shielded, though the requirements differ.

In a full/high-speed cable, an aluminum metallized polyester shield surrounds the four conductors. Around this shield is an outer shield of braided, tinned copper wire. Between the shields and contacting both is a copper drain wire. The outside layer is a polyvinyl chloride jacket. The shield terminates at the connector plug.

A low-speed cable has the same requirements except that the braided outer shield is recommended but not required. The 1.x specification required no shielding for low-speed cables on the premise that the slower rise and fall times made shielding unnecessary. The shielding requirement was added in USB 2.0 not because the USB interface is noisy in itself, but because the cables are likely to attach to computers that are noisy internally. Shielding helps to keep the cable from radiating this noise and helps the cable pass FCC tests. The downside is that 2.0-compliant low-speed cables are more expensive to make and physically less flexible.

Edge Rates

Low speed's slower data rate enables the drivers to use slower edge rates that reduce both the reflected voltages seen by receivers and the noise that emanates from the cable.

When a digital output switches, a mismatch between the line's characteristic impedance and the load presented by the receiver can cause reflected voltages that affect the voltage at the receiver. If the reflections are large enough and last long enough, the receiver may misread a transmitted bit.

In low-speed cables, the slower edge rate ensures that any reflections have died out by the time the output has finished switching. The slow edge rate also means that the signals contain less high-frequency energy and thus the noise emanated by the cables is less.

Isolated Interfaces

Galvanic isolation can be useful in preventing electrical noise and power surges from coupling into a circuit. Circuits that are galvanically isolated from each other have no ohmic connection. Typical methods of isolation include using a transformer that transfers power by magnetic coupling and optoisolators that transfer digital signals by optical coupling.

USB devices should require no additional protection in offices, classrooms, and similar environments. For industrial environments or anywhere that devices require additional protection, USB's timing requirements and use of a single pair of wires for both directions make it difficult to completely isolate a USB device from its host. It is feasible, however, to isolate the circuits that a device controller connects to. For example, in a motor controller with a USB interface, the motor and control circuits can be isolated from the USB controller and bus.

Another option is an isolated hub available from B & B Electronics. The hub has four low- and full-speed downstream ports with 2500 VAC of optical isolation between the upstream port and the downstream ports.

Wireless Links

For the same reasons that isolated USB interfaces are difficult to implement, replacing a USB cable with a wireless connection isn't a simple task. USB transactions involve communicating in both directions with tight timing requirements. For example, when a host sends a token and data packet in the Data stage of an interrupt OUT transaction, the device must respond quickly with ACK or another code in the handshake packet.

But the idea of a wireless connection for USB devices is so appealing that several technologies that incorporate USB in wireless devices are available and under development. In most implementations, the wireless links use conventional wired devices that serve as wireless bridges, or adapters. The bridge or adapter uses USB to communicate with the host and a wireless link to communicate with the peripheral. The peripheral contains a wireless bridge to convert between the wireless interface and the peripheral's circuits.

Cypress WirelessUSB

Cypress Semiconductor offers the WirelessUSB technology as a solution for low-speed devices, including HIDs, without cables. The obvious market is wireless keyboards, mice, and game controllers. With a wireless range of up to 50 meters, the technology might also find uses in building and home automation and industrial control. The wireless interface uses radio-frequency (RF) transmissions at 2.4 Gigahertz in the unlicensed Industrial, Scientific, and Medical (ISM) band.

A WirelessUSB system consists of a WirelessUSB bridge and one or more WirelessUSB devices (Figure 19-10). The bridge translates between USB and the wireless protocol and medium. The WirelessUSB device carries out the device's function (mouse, keyboard, game controller) and communicates with the bridge.

The bridge contains a USB-capable microcontroller and a WirelessUSB transceiver chip and antenna. The WirelessUSB device contains a Cypress PsOC or another microcontroller and a WirelessUSB transmitter or transceiver chip and antenna. A device with a transceiver is 2-way: the device can communicate in both directions. A device with just a transmitter is 1-way: the device can send data to the host but can't receive data or status information. In both the bridge and device, the transmitter and transceiver chips use the SPI synchronous serial interface to communicate with their microcontrollers.

In a 2-way system, when a device has data to send to the host, the device's microcontroller writes the data to the transceiver chip, which encodes the data and transmits it through the air to the bridge's transceiver. On receiving the data, the bridge returns an acknowledgement to the device, decodes the data, and sends the data to the host in conventional USB interrupt or control transfers. If the device doesn't receive an acknowledgement from the bridge, the device resends the data.

When the host has data to send to the device, the host writes the data to the bridge's USB controller, which ACKs the data (if not busy) and passes the data to the bridge's transceiver. The transceiver encodes the data and sends it over the air to the WirelessUSB device. The device returns an acknowledgement to the bridge. On receiving a NAK or no reply, the bridge retries the transmission.

In a 1-way system, a device sends data to the host in much the same way as in a 2-way system, except that the device receives no acknowledgements from the host. To help ensure that the bridge and host receive all transmitted data, the device sends its data multiple times. Sequence numbers enable the bridge to identify previously received data.

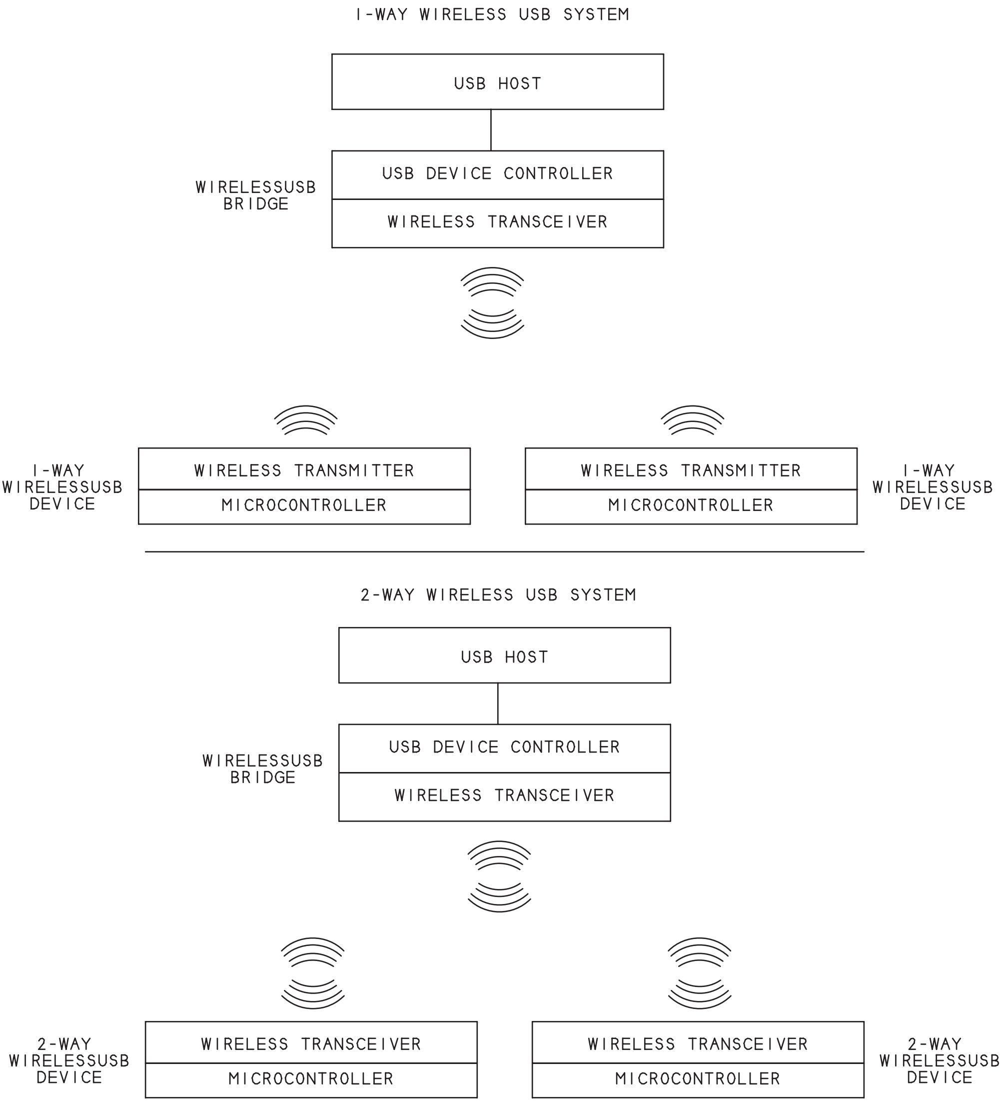

Figure 19-10: WirelessUSB provides a way to design low-speed devices that use a wireless interface.

With both systems, the host thinks it's communicating with an ordinary HID and has no knowledge of the wireless link.

A WirelessUSB link can have a data throughput of up to 62.5 kilobits/sec., but low-speed traffic is of course limited to the USB bandwidth available for low-speed control and interrupt transfers. A device and its bridge must use the same frequency/code pair. A single WirelessUSB bridge can use multiple frequency/code pairs to communicate with multiple devices.

For devices with human interfaces, communications between the wired and wireless interfaces must be fast enough so users don't perceive delays in response to keypresses, mouse movements, and similar actions. For faster performance, the microcontroller can use burst reads to read multiple registers in the WirelessUSB chip in sequence.

The Wireless USB Initiative

The mission of the Wireless USB Promoter Group is to specify a Wireless USB (WUSB) extension that can transmit at 480 Megabits/sec. over a distance of 3 meters (and at lower speeds over longer distances). Note that Wireless USB (WUSB) and Cypress' WirelessUSB have similar names but are different and unrelated technologies!

In Wireless USB, a conventional USB host can have a wired connection to a USB device that functions as a host wire adapter (HWA). The HWA can communicate with native WUSB devices and with device wire adapters (DWAs). A native WUSB device is a peripheral with Wireless USB support built in. A DWA connects to a conventional wired USB device and enables the wired device to communicate over the wireless link. Data on the wireless link is encrypted.

The members of the Wireless USB Promoter Group are Agere Systems, Hewlett Packard, Intel, Microsoft Corporation, NEC, Philips Semiconductors and Samsung Electronics. The specification is due for release in 2005.

Other Options

Other ways to use USB in wireless devices include various wireless bridges and a wireless networking option.

ZigBee is an inexpensive, low-power, RF interface suitable for building and industrial automation and other applications that transmit at up to 250 kilobits/sec. and over distances of up to 500 meters. DLP Design's DLP-RF1 USB/RF OEM Transceiver Module provides a way to monitor and control a Zigbee interface from a USB port. The module's USB controller is FTDI Chip's FT245BM. One or more DLP-RF2 RF OEM Transceiver Modules can communicate with the DLP-RF1.

The IrDA Bridge class described in Chapter 7 defines a way for a USB device to use bulk transfers to communicate over an infrared link.

Another option is a vendor-specific wireless bridge that uses infrared, RF, or other wireless modules designed for use in robotics and other low- to moderate-speed applications. The bridge functions as a wired USB device and supports a wireless interface. A remote device carries out the peripheral's function and also supports the wireless interface. Firmware in the bridge passes received wireless data to the host and passes received USB data to the device.

If you want to use an existing USB device wirelessly, you may be able to use the AnywhereUSB or Keyspan hub/server described earlier in this chapter with a wireless network interface between the host PC and the hub/server.