

Index

Page numbers in italics indicate figures and tables.

A

- ACK (acknowledge) handshake code, 48, 52, 53, 55, 56, 56, 57
- ACPI (Advanced Configuration and Power Interface) specification, 467
- Address field, 500, 500
- Advanced Configuration and Power Interface (ACPI) specification, 467
- Agere Systems, 166, 533
- AMD chips, 155
- AnywhereUSB, 525, 534
- API (Applications Programming Interface). *See* Windows API functions
- Apple Macintosh, support for USB, 7–8
- Applications Programming Interface (API). *See* Windows API functions
- assemblers, as necessary tool for USB device development, 30
- Atmel chips, 155
- audio class
 - about, 181–2, 182
 - chips, 188
 - descriptors, 182–8
 - documentation, 181
 - INF file, 276
 - types of Unit descriptor, 184, 187
 - where declared, 179
 - Windows support, 189
- AudioControl interface, 183, 183
- AudioStreaming interface, 184, 185

B

- B & B Electronics, 530
- babble, 444

- bandwidth
 - for bulk transfers, 71
 - for control transfers, 66, 67, 321
 - for interrupt transfers, 72, 74, 75, 81
 - for isochronous transfers, 78, 79, 80, 81
 - role in data transfer, 24
 - role of translation translator, 71
 - as speed-limiting factor, 81
- batteries. *See* self-powered devices
- Battery Devices device setup class, 251
- BIOS, 10–11
- bit stuffing, 496–7
- bridges, wireless, 534
- broadcasting, 11
- buffers, in USB controllers, 143–4
- bulk endpoints, 54, 58
- bulk transfers
 - about, 41–2, 46, 68
 - asynchronous serial interface example, 408, 409, 410, 411–13, 414
 - availability, 69
 - vs. control transfers, 41, 46
 - data size, 69, 71
 - data toggles in, 58
 - error handling, 71–2
 - vs. interrupt transfers, 41, 46
 - vs. isochronous transfers, 41, 46
 - maximum data-transfer rate per endpoint, 62
 - parallel interface example, 414, 415, 416, 417, 418–21, 421
 - speed, 71
 - and stream pipes, 43
 - structure, 69, 70
 - transactions, 36, 46
- bus drivers, 234, 237, 238, 239, 239, 240, 246, 257
- bus power
 - about, 24, 26–7, 455–6

Index

- and hubs, 434, 461–2, 463
 - vs.* self-power, 26–7, 457, 460, 462
 - soft-start capability, 466
 - voltage issues, 456–7
 - which peripherals can use, 457, 458
- bus speed
- about, 5
 - and bulk transfers, 71
 - and control transfers, 66–8
 - converting, 50–1
 - and device controllers, 28–9
 - FTDI Chip performance tips, 426–9
 - host computer support, 36–7, 37
 - and interrupt transfers, 73–5
 - and isochronous transfers, 79–80
 - low *vs.* high, 28–9, 50–1
 - maximum data-transfer rate per endpoint, 62
 - and signal voltage, 517–18, 518
 - and split transactions, 50–1, 439–44, 441, 442
 - switching, 516
 - USB 1.x *vs.* USB 2.0, 5, 14–16
 - USB limitations, 10
 - USB *vs.* other interfaces, 3, 10
- bus states
- about, 489
 - defined, 489
 - full-speed, 490–2
 - high-speed, 492–4
 - low-speed, 490–2
 - signal voltages, 517–18
- ByRef keyword, 289
- ByVal keyword, 289
- ## C
- C++. *See* Visual C++ .NET
- cables
- about, 518–20, 520
 - and bus power, 456–7
 - captive, 524
 - conductors, 518–20
 - connectors, 520–3, 521, 522, 523
 - detachable, 524
 - extension, 525
 - full-speed segments, 506–8, 507, 519, 520
 - high-speed segments, 506–8, 507, 519, 520
 - length issues, 10, 447–8, 524–5
 - low-speed segments, 507, 520
 - for On-The-Go (OTG) devices, 542, 543, 544
 - vs.* other interfaces, 3, 10
 - plug types, 521, 521, 522
 - requirements, 518–25
 - twisted-pair, 527–8, 528
 - in USB 2.0 specification, 518–25
- captive cables, 524
- catalog (.cat) files, 274, 486
- Catalyst Enterprises, Inc., 477
- CATC USBTracer/Trainer, 477
- CBW (Command Block Wrapper), 209–10, 210
- CCIDs (Chip Card Interface Devices), 189–91
- CD-ROM Drives device setup class, 251
- Centronics, 11, 13
- Certified USB logo, 477, 484, 485. *See also* USB icon
- checklists, compliance, 478–9
- Chip Card Interface Devices (CCIDs), 189–91
- chip cards, 189. *See also* chip/smart card interface class
- chips, controller
- about, 8–9
 - compatibility list, 155
 - Cypress CY7C67200 EZ-OTG, 548–9
 - Cypress enCoRe, 155, 163
 - Cypress EZ-USB family, 155, 157–62, 159
 - Cypress WirelessUSB, 530–1, 532, 533
 - FDTI FT232BM, 166, 170, 171, 173, 174–6, 227–9, 228
 - FDTI FT245BM, 166, 170, 171, 172, 174–6
 - Freescale MC68HC08 family, 164

- Freescale MCF5482 ColdFire, 164
- Microchip Technology PIC16C745/65, 155, 157
- Microchip Technology PIC16F877 microcontroller in FT232BM device controller example, 408, 409, 410, 411–13, 414
- Microchip Technology PIC16F877 microcontroller in FT245BM device controller example, 414, 415, 416, 417, 418–21, 421
- Microchip Technology PIC18F4550, 155, 156–7, 338
- National Semiconductor USBN9603, 165–7
 - for OTG devices, 545–9
 - Philips ISP1181B, 166, 167–8
 - Philips ISP1261, 549
 - Philips ISP1362, 546–7
 - Philips ISP1581, 166, 168–9
 - Philips PDIUSB12, 166
 - PLX Technology NET2272 controller, 166, 169–70
 - TransDimension TD242LP, 547–8
- chip/smart card interface class, 179, 189–91, 251, 276
- Chirp J bus state, 493
- Chirp K bus state, 493
- ChkInf tool, 266, 277
- class descriptor, HID, 326–8, 327
- class drivers
 - advantages of using, 226–7
 - vs.* custom drivers, 226–7
 - as developer challenge, 12–13
 - INF files, 276
 - operating system support, 8
- classes
 - about, 177–80
 - audio class, 179, 181–9
 - chip/smart card interface class, 179, 189–91, 251, 276
 - communication class, 179, 191–8, 251
 - Content Security class, 179, 198–200
 - Device Firmware Upgrade (DFU) class, 179, 200–2
 - device setup, 250–1
 - and Device Working Groups, 178
 - Human Interface Device (HID) class, 179, 203–6, 251, 276, 319
 - IrDA Bridge class, 179, 206–8, 534
 - mass-storage class, 179, 208–13, 251
 - printer class, 179, 213–17, 251
 - specification elements, 178–80, 180
 - standard *vs.* custom drivers, 226–7
 - still-image class, 179, 217–20, 251
 - test-and-measurement class, 179, 220–1
 - video class, 179, 221–6
- class installers, 279
- class keys, Windows registry, 250, 259–60, 260
- class specifications, 178–80, 179, 180
- class-specific requests
 - about, 140
 - audio class, 188
 - chip/smart card interface, 191
 - communication-device class, 196
 - Content Security class, 199–200
 - Device firmware Upgrade class, 202
 - HIDs, 204–5
 - hub class, 451, 452–3
 - IrDA interface, 207
 - mass-storage devices, 212
 - printer class, 215
 - still-image class, 219
 - test-and-measurement class, 221
 - video class, 225
- Clear_Feature request, 128, 130, 452
- Clear_Hub_Feature request, 451, 452, 453
- Clear_Port_Feature request, 451, 452, 454
- Clear_TT_Buffer request, 451, 452
- client drivers, in layered driver model, 238–40, 239, 240, 245
- clocks, in USB controllers, 145
- CloseHandle API function
 - about, 306
 - defined, 304
 - in Visual Basic code, 306
 - in Visual C++ code, 306
- CLR (common language runtime), 282
- code, managed *vs.* unmanaged, 282–4
- co-installers, 279
- Collection items, 360–1, 361
- Command Block Wrapper (CBW), 209–10, 210

Index

- Command Status Wrapper (CSW), 210–11, 211
- common language runtime (CLR), 282
 - communication-device class
 - about, 192–3
 - chips, 196–7
 - descriptors, 193, 193–6, 195, 196, 197
 - device setup class equivalent, 251
 - documentation, 192
 - where declared, 179
 - Windows support, 197–8
- Compaq
 - and OHCI standard, 241
 - and USB 2.0 specification, 14
- compatible IDs, 272, 273, 274, 275
- compilers, as necessary tool for USB device development, 30
- compliance testing, 478–84, 479, 480, 483
- composite devices
 - defined, 21, 103–4
 - in enumeration process, 90, 91
 - finding drivers, 274–5
 - identification strings, 272–3
 - layered drivers for, 239
- compound devices
 - about, 21
 - bus-powered, 461
 - defined, 21
 - hubs in, 434, 450
 - in peripheral-host PC configurations, 19, 20
 - speed issue, 28
- computer ports. *See* ports
- conductors, 518–20
- configuration descriptor, 94, 95, 101–3, 102
- Connect bus state, 492
- connectors, 520–3, 521, 522, 523
 - for On-The-Go (OTG) devices, 542, 543, 544
- Content Security class, 179, 198–200
- control endpoints
 - about, 38
 - for HID transfers, 321–2
 - number of, 62, 66
 - support for functional stall, 54
- controllers. *See* chips, controller; device controllers; host controllers
- Control Read requests, 126–7
- control transfers
 - about, 40, 41, 46, 61
 - availability, 62
 - vs. bulk transfers, 41, 46
 - data size, 66
 - Data stage, 46, 62, 63, 64, 65, 120–2
 - data toggles in, 58
 - elements, 117–27
 - error handling, 68, 124–5
 - in HID transfers, 321, 322, 322
 - vs. interrupt transfers, 41, 46
 - vs. isochronous transfers, 41, 46
 - maximum data-transfer rate per endpoint, 62
 - and message pipes, 42
 - receiving HID reports via, 347–8, 349
 - requests, 127, 128, 129–39
 - sending HID reports via, 345, 346, 347, 348
 - Setup stage, 46, 62, 63, 64, 65, 118–20
 - speed, 66–8
 - status codes, 55
 - Status stage, 46, 62, 63, 64, 65, 122–4
 - structure, 62–5
 - transactions, 36, 46
- Control Write requests, 125–6
- cost, USB, 6
- CPUs
 - embedded, controllers with, 154–64
 - in USB devices, 145
- CRC (cyclic redundancy check), 56–7
- CRC field, 500, 501
- CreateFile API function
 - about, 303, 305–6
 - defined, 304
 - in Visual Basic .NET code, 304–5
 - in Visual C++ code, 303–4
- crystals, 498–9
- CSW (Command Status Wrapper), 210–11, 211
- cyclic redundancy check (CRC), 56–7
- Cypress Semiconductor
 - CY7C67200 EZ-OTG controller chip, 548–9
 - enCoRe, 155, 163
 - EZ-USB chip family, 155, 157–62, 159

WirelessUSB, 530–1, 532, 533

D

- D2XX Direct driver, 421, 422, 423–4, 424–6, 425
- data encoding. *See* Non-Return to Zero Inverted (NRZI) with bit stuffing encoding
- Data field, 500, 501
- data flow
 - direction codes, 39, 41, 45
 - role of host, 24
- Data J bus state, 490–1. *See also* High-speed Data J bus state
- Data K bus state, 490–1. *See also* High-speed Data K bus state
- data memory, 147
- data packets, 44, 44, 45, 46, 47, 49–50, 52, 53, 118–20, 121, 123
- data phase, transactions. *See* data packets
- Data stage
 - about, 46, 120–1
 - in control transfer structure, 62, 63, 64, 65
 - Control Write requests, 125–6
 - data packets, 121
 - handshake packets, 122
 - token packets, 121
- data toggles, 57–9
- data types, Visual Basic *vs.* .NET Framework, 288
- DDK (Driver Development Kit), 30
- debug descriptor, 95, 113
- debugging tools, 30, 151–2
- Default Control Pipe, 40, 62, 66
- Delimiter item, 371, 372
- descriptors
 - about, 93
 - audio class, 182–8
 - chip/smart card interface class, 190–1
 - class-specific, 94–5
 - communication-device class, 193, 193–6, 195, 196, 197
 - configuration, 94, 95, 101–3, 102
 - debug, 95, 113
 - defined, 93
 - device, 94, 95, 96–9, 97, 99
 - device_qualifier, 94, 95, 99–101, 100
 - endpoint, 94, 95, 108–12, 110
 - in enumeration process, 89–90
 - HID class, 326–8, 327
 - interface, 94, 95, 106–8, 108, 109
 - interface association, 95, 103–6, 106
 - interface_power, 95, 113
 - making 2.0-compliant, 114–15, 115
 - Microsoft OS, 114
 - OTG, 95, 113, 545, 546
 - other_speed_configuration, 94, 95, 103, 104
 - padding, 373
 - physical, 373
 - report, 328–30, 329
 - string, 94, 95, 112–13, 113
 - types, 94–5
 - vendor-specific, 94–5, 324, 325
 - video class, 223–5
- Designator Index item, 371, 372
- Designator Maximum item, 371, 372
- Designator Minimum item, 371, 372
- detachable cables, 524
- DeVaSys Embedded Systems USB 12C/IO board, 152–3, 153
- developers
 - USB benefits, 6–9
 - USB challenges, 12–13
 - USB limitations, 10–12
- development kits
 - from chip manufacturers, 151–2
 - as necessary tool for USB device development, 30
 - printed-circuit board alternatives, 152–4
- device controllers. *See also* chips, controller and bus speed, 28–9
 - chip documentation, 150
 - debugging tools, 151–4
 - development overview, 148–54
 - driver choices, 151
 - elements, 142–8
 - with embedded CPUs, 154–64
 - with interface to external CPU, 165–76, 166
 - as necessary element for USB devices, 29
 - requirements, 149–50

Index

- selecting, 141–76
 - USB controller overview, 143–5
 - what they do, 8–9, 25, 27
- device descriptor, 94, 95, 96–9, 97, 99
- device drivers
 - about, 233–7
 - creating by using driver toolkit, 248–9
 - custom, creating, 247–9
 - defined, 233
 - digitally signed, 486–7
 - in enumeration process, 90–3
 - generic, 231, 248
 - layered driver model, 234, 235, 236, 238–42, 239
 - making initial decision, 31
 - matching to devices, 253–80
 - as necessary element for USB devices, 30
 - role in data transfer, 235–7
 - selecting, 279–80, 422, 423–4, 424–6
 - as speed-limiting factor, 82–3
 - user mode *vs.* kernel mode, 235–7, 249
 - viewing information in Device Manager, 254, 254–7, 255, 256
 - ways to obtain, 235
 - Windows search process, 279–80
 - writing, 30, 247
- Device Firmware Upgrade (DFU) class, 179, 200–2
- Device Framework tests, 479–81
- device identification strings
 - finding matches, 274–5
 - obtained from devices, 272–3
 - obtained from INF files, 273
- device IDs, 272–3, 274, 275
- device installers, 279
- device interface classes
 - obtaining GUIDs, 291, 292–3
 - registering for device notifications, 307
 - requesting pointers to, 293–5
 - role of GUIDs, 249, 250, 251
- device interface GUIDs
 - about, 251–2
 - defined, 249
 - vs.* device setup GUIDs, 249, 252
 - obtaining, in Visual Basic code, 293
 - obtaining, in Visual C++ code, 292
- device interfaces
 - extracting device path names, 301–2
 - identifying, 295–8
 - retrieving device path names, 298–301
- Device Manager
 - about, 253
 - enabling/disabling remote wakeup capability, 468
 - viewing device information in, 254, 254–7, 255, 256
 - viewing Properties pages, 257, 279
- device_qualifier descriptor, 94, 95, 99–101, 100
- devices, USB
 - about, 21
 - bus-powered *vs.* self-powered, 26–7, 457, 460, 462
 - closing communication handles, 306
 - composite, 21, 90, 91, 103–4, 239, 272–3, 274–5
 - compound, 19, 20, 21, 28, 434, 450, 461
 - controllers. *See* device controllers
 - defined, 21
 - detecting, 281–318
 - detecting attachment and removal, 306–18
 - developing, 29–32
 - elements, 29–30
 - enumeration process, 86–93
 - identifying as HID, 323–30
 - low-power *vs.* high power, 458–60
 - low- *vs.* high-speed, 28–9
 - preparing to enumerate, 31–2
 - programming applications to find, 291–303
 - removing, 92
 - requesting communication handles, 303–6
 - resuming communication, 466
 - role of host, 23
 - specifying requirements, 31
 - speed-limiting factors, 81–2
 - steps in project development, 30–2
 - suspended, 464–6
 - testing, 477–87
 - tools for developing, 30

- tools for testing and debugging, 471–7
- viewing information in Device Manager, 254, 254–7, 255, 256
- in Windows registry, 258–9
- wireless connections, 530–4
- device setup classes, 249, 250, 251
- device setup GUIDs
 - about, 250–1
 - defined, 249
 - vs.* device interface GUIDs, 249, 252
 - in Windows registry, 259, 260, 267
- Device Working Groups, 178
- Differential 1 bus state, 490. *See also* High-speed Differential 1 bus state
- Differential 2 bus state, 490. *See also* High-speed Differential 2 bus state
- digital cameras, 457
- digital signatures, 486–7
- Disconnect bus state, 492. *See also* High-speed Disconnect bus state
- Disk Drives device setup class, 251
- distance
 - USB limitations, 10
 - USB *vs.* other interfaces, 3, 10
- DLL files, 279, 284
- DLP Design, 166, 534
- DOS, 11
- Driver Development Kit (DDK), 30
- driver keys, Windows registry, 260–2, 261, 279
- drivers. *See* class drivers; device drivers
- Driver X USB toolkit, 231
- dual-speed devices, detecting speed, 116

E

- edge rates, 529
- EEPROM (electrically erasable PROM)
 - memory
 - about, 146–7
 - and Cypress Semiconductor EZ-USB chip family, 160–2, 161
 - programming, 429, 430–1, 432
- EHCI (Enhanced Host Controller Interface), 239, 241, 242

- EIA/TIA-232 interface *vs.* USB, 3
- Ellisys USB Explorer protocol analyzer, 263, 264–5, 473, 474, 474–5, 475
- End Collection items, 360–1, 361
- End-of-High-speed-Packet (HSEOP) bus state, 494
- End-of-Packet (EOP) bus state, 491–2, 498
- End-of-Packet (EOP) signal
 - about, 48–9
 - and Resume state, 466
 - and SYNC field, 497–8
- endpoint descriptor, 94, 95, 108–12, 110
- Endpoint field, 500, 500
- endpoints
 - about, 38–40
 - addresses, 38–9
 - defined, 38
 - for HID transfers, 321–2
- Enhanced Host Controller Interface (EHCI), 239, 241, 242
- enumeration
 - about, 86–7
 - as aspect of USB transfer, 34
 - defined, 85
 - power issues, 459–60
 - preparing for, 31–2
 - steps in process, 87–91
 - tips for success, 92–3
- EOP (End-of-Packet) signal, 48–9
- EPROM (erasable programmable ROM)
 - memory, 146
- ERR handshake code, 52, 53, 55
- error handling
 - in bulk transfers, 71–2
 - in control transfers, 68
 - in interrupt transfers, 75–6
 - in isochronous transfers, 80
 - role of data toggles, 57–9
 - role of host, 24
 - role of peripherals, 26
 - in USB transfers, 56–9
- ERR PID, 48, 49
- Ethernet *vs.* USB, 3, 10, 17
- extension cables, 525
- EZ-USB boards, 152

F

Feature items, 356, 357, 358–60
 Feature reports
 Main item types in, 356, 357, 358–60
 reading from devices, 404–5
 transfer types, 389, 389
 writing to devices, 402–4
 fields, packet, 499–501, 500
 FireWire 800 *vs.* USB, 3
 firmware, defined, 146. *See also* device drivers
 flash memory, 146
 480 Megabits/sec, 512
 Frame Number field, 500, 501
 frames
 in bulk transfers, 71
 in control transfers, 66, 67
 defined, 24, 35
 and frame numbers, 48, 501
 illustrated, 35
 in interrupt transfers, 74–5
 IrDA, 206, 207
 in isochronous transfers, 76–80
 role in USB transfer, 35–6
 in split transactions, 443
 and timing accuracy, 499
 in transactions, 48, 50, 66
 Freescale Semiconductor
 68HC05JB3/4 chip, 155
 68HC08JB8 chip, 155
 MC68HC08 family, 164
 MCF5482 ColdFire, 155, 164
 FTDI. *See* Future Technology Devices International (FTDI)
 full speed, defined, 5
 full-speed bus states, 490–2
 full-speed transceivers, 508–12
 functions, defined, 19, 21
 Future Technology Devices International (FTDI)
 D2XX Direct driver, 421, 422, 423–4, 424–6, 425
 FT232BM, 166, 170, 171, 173, 174–6, 227–9, 228
 in bulk transfer example, 408, 409, 410, 411–13, 414

FT245BM, 166, 170, 171, 172, 174–6
 in bulk example, 414, 415, 416, 417, 418–21, 421
 improving chip performance, 426–9
 Virtual COM Port Driver, 174, 227, 421

G

galvanic isolation, 529–30
 generic drivers, 231, 239, 248
 GenInf tool, 277
 Get_Bus_State request, 451, 452
 Get_Configuration request, 128, 135, 452
 Get_Descriptor request, 94, 128, 133, 452
 Get_Hub_Descriptor request, 451
 Get_Hub_Status request, 451, 452, 453
 Get_Idle request, 331, 333
 Get_Interface request, 128, 137, 452
 Get_Port_Status request, 444, 451, 452, 454
 Get_Protocol request, 331, 334
 Get_Report request, 331, 332
 Get_Status request, 128, 129, 444, 452, 460
 Get_TT_State request, 451, 452
 Global item type
 defined, 361
 Logical Maximum item, 362, 364–5
 Logical Minimum item, 362, 364–5
 Physical Maximum item, 362, 365, 366
 Physical Minimum item, 362, 365, 366
 Pop item, 362, 370
 Push item, 362, 369
 Report Count item, 362, 369
 Report ID item, 361–3, 362
 Report Size item, 362, 369
 Reserved item, 362
 Unit Exponent item, 365, 366
 Unit item, 362, 366
 Usage Page item, 362, 363–4
 Globally Unique Identifiers. *See* GUIDs (Globally Unique Identifiers)
 Gold Tree testing configuration, 482, 483, 484
 GPIB interface *vs.* USB, 3

GUIDs (Globally Unique Identifiers)
 about, 249–50
 device interface, 249, 251–2, 292, 293
 device setup, 249, 250–1, 252, 259, 260,
 267
 in Windows registry, 259, 260, 267

H

handles, communication

closing, 306
 opening, 303–6

handshake packets, 44, 45, 46, 47, 48, 50,
 52, 53, 120, 122, 123–4

handshake phase, transactions. *See* handshake
 packets

handshaking

about, 51–2
 hardware *vs.* software, 51–2
 status codes, 52–5, 53

hardware IDs, 272, 274

hardware keys, Windows registry, 258,
 258–9, 261, 279–80

hardware protocol analyzers, 472, 472–5,
 474, 475

Hewlett-Packard

and USB 2.0 specification, 14
 and Wireless USB Promoter Group, 533

HID Descriptor Tool, 352, 353

HidD_FlashQueue API function, 379, 381

HidD_FreePreparedData API function,
 377, 406

HidD_GetAttributes API function

about, 384
 in Visual Basic code, 382–3
 in Visual C++ code, 381–2

HidD_GetFeature API function

about, 378, 404, 405
 defined, 379
 Visual Basic in, 405
 Visual C++ in, 404

HidD_GetHidGuid API function

defined, 292, 381
 in Visual Basic code, 293
 in Visual C++ code, 292

HidD_GetIndexedString API function, 378

HidD_GetInputReport API function, 378,
 379, 405

HidD_GetManufacturerString API func-
 tion, 378

HidD_GetNumInputBuffers API function,
 381

HidD_GetPhysicalDescriptor API function,
 377

HidD_GetPreparedData API function, 376,
 377, 388, 406

about, 384, 385
 in Visual Basic code, 385
 in Visual C++ code, 384

HidD_GetProductString API function, 378

HidD_GetSerialNumberString API func-
 tion, 378

HidD_SetFeature API function

about, 378, 402, 403–4
 defined, 379
 in Visual Basic code, 403
 in Visual C++ code, 402

HidD_SetNumInputBuffers API function,
 379, 381, 400

HidD_SetOutputReport API function, 378,
 379

HidP_GetButtonCaps API function, 376,
 377, 388

HidP_GetButtons API function, 379, 380

HidP_GetButtonsEx API function, 380

HidP_GetCaps API function

about, 385, 388
 defined, 377
 reading feature reports from devices, 405
 reading input reports from devices, 393,
 400

requesting HID information, 376

in Visual Basic code, 387

in Visual C++ code, 386

writing feature reports to devices, 403

writing output reports to devices, 392

HidP_GetData API function, 380

HidP_GetExtendedAttributes API function,
 377

HidP_GetLinkCollectionNodes API func-
 tion, 377

HidP_GetScaleUsageValue API function,
 380

Index

- HidP_GetSpecificButtonCaps API function, 377
- HidP_GetSpecificValueCaps API function, 377
- HidP_GetUsages API function, 380
- HidP_GetUsagesEx API function, 380
- HidP_GetUsageValue API function, 379, 380
- HidP_GetUsageValueArray API function, 380
- HidP_GetValueCaps API function, 377, 388
- HidP_InitializeReportForID API function, 380
- HidP_IsSameUsageAndPage API function, 377
- HidP_MaxDataListLength API function, 377
- HidP_MaxUsageListLength API function, 377
- HidP_SetButtons API function, 379, 380
- HidP_SetData API function, 380
- HidP_SetScaledUsageValue API function, 380
- HidP_SetUsages API function, 380
- HidP_SetUsageValue API function, 379, 380
- HidP_SetUsageValueArray API function, 380
- HidP_TranslateUsagesToI8042ScanCodes API function, 377
- HidP_Unsetbuttons API function, 379, 380
- HidP_UnsetUsages API function, 380
- HidP_UsageAndPageListDifference API function, 377
- HidP_UsageListDifference API function, 377
- HidRegisterMinidriver API function, 381
- HIDs. *See also* Human Interface Device (HID) class; reports, HID
 - abilities, 320–1
 - about, 320–1
 - API function overview, 375–9
 - API functions for managing communications, 379, 381
 - class descriptor, 326–8, 327
 - descriptor example, 323, 324–5
 - designating devices as, 226–7
 - firmware requirements, 323
 - hardware requirements, 321–3
 - identifying devices as, 323, 324–5, 325–30, 327, 329
 - limitations, 320–1
 - list of requests, 331, 332–7
 - receiving reports via control transfers, 347–8, 349
 - receiving reports via interrupt transfers, 343, 344, 345
 - report descriptors, 328–30
 - requesting information about, 376, 377, 378
 - role of human interface in, 320
 - sending reports via control transfers, 345, 346, 347, 348
 - sending reports via interrupt transfers, 340, 341, 342
 - vendor-specific example, 323, 324–5, 325
- HID Usage Tables* document, 354
- high-speed, defined, 5, 15–16
- high-speed bus states, 492–4
 - High-speed Data J bus state, 493
 - High-speed Data K bus state, 493
 - High-speed Differential 1 bus state, 492
 - High-speed Differential 2 bus state, 492
 - High-speed Disconnect bus state, 494
 - High-speed Idle bus state, 494
 - High-speed Squelch bus state, 494
- high-speed transceivers, 512–17
- HNP (Host Negotiation Protocol), 539–41, 545
- host computers
 - about, 17
 - bus speed support, 36–7
 - configurations for connecting USB devices, 19, 20, 29
 - defined, 17
 - enumeration process, 86–93
 - hardware and software requirements, 22–3
 - PC-to-PC communication, 229–30, 230
 - power issues, 24–5, 460, 460, 462, 463, 467–9, 468
 - speed-limiting factors, 82–4

- support as necessary element for USB devices, 29
- transfer responsibilities, 35–6
- what they do, 22–5
- host-controller drivers
 - about, 23, 240
 - in layered driver model, 239, 241–2
 - from Microsoft, 479
 - for OHCI controller type, 241–2
 - role in data communication flow, 245, 246
 - from Transdimension, 548
 - for UHCI controller type, 241–2
- host controllers, 17, 19, 22, 23
- Host Negotiation Protocol (HNP), 539–41, 545
- host wire adapter (HWA), 533
- hub class, 178, 179, 276, 448–54, 451
- hub-class descriptor, 449–50
- hub-class requests, 444, 451, 451–2, 453
- hub controllers, 444–5
- hub drivers, 239, 240, 246, 272, 275. *See also* bus drivers
- hubs. *See also* root hub
 - 1.0 descriptors, 448
 - 2.0 descriptors, 448
 - about, 21, 433–4, 434
 - bus-powered, 434, 461–2, 463
 - cable-length issues, 447–8
 - defined, 21
 - enumeration process, 91
 - low-power, 459–60, 461
 - over-current condition, 462–3
 - power options, 461–3
 - power switching support, 463
 - role of hub controller, 444–5
 - role of repeaters, 435–8, 436
 - role of transaction translators, 438–44, 439
 - self-powered, 462
 - in series, 447–8
 - speed issue, 445–7, 446
 - in tiered-star topology, 18, 18–19
- Human Interface Device (HID) class, 179, 203–6, 251, 276, 319. *See also* HIDs
- HWA (host wire adapter), 533

- I²C interface *vs.* USB, 3
- icon, USB, 522, 523. *See also* logo, USB
- Idle bus state, 491. *See also* High-speed Idle bus state
- IEEE-488 interface *vs.* USB, 3
- IEEE-1394a interface *vs.* USB, 10
- IEEE-1394b interface *vs.* USB, 3, 10
- IEEE-1394 interface *vs.* USB, 16–17
- Imaging Device device setup class, 251
- INF files
 - about, 262–3
 - for classes of devices, 276–7
 - ClassInstall32 section, 267–8
 - copyright comment, 266
 - deleting, 278–9
 - DestinationDirs section, 269
 - filename conventions, 279
 - finding, 278
 - Install section, 270–1
 - list of sections, 266–71
 - manufacturer section, 268
 - Models section, 269–70
 - sample listing, 264–5
 - searching for device identification strings, 272–7
 - SourceDiskFiles section, 269
 - SourceDiskNames section, 269
 - Strings section, 271
 - syntax, 265–6
 - tools and diagnostic aids, 277–80
 - version section, 266–7
- Infineon, 155
- Input items, 356, 357, 358–60
- Input reports
 - Main item types in, 356, 357, 358–60
 - reading from devices, 392–401
 - transfer types, 389, 389
- Intel Corporation
 - 80C186 chips, 155
 - 8051 chips, 9, 145, 154, 155, 158, 159, 159, 160, 162, 175
 - and UHCI standard, 241
 - and USB 2.0 specification, 14
 - and Wireless USB Promoter Group, 533

Index

interface association descriptor, 95, 103–6, 106

interface descriptor, 94, 95, 106–8, 108, 109

interface_power descriptor, 95, 113

interoperability tests, 481–4

interrupt endpoints

- bandwidth issues, 81
- and data toggle, 58
- in enumeration process, 87
- for HID transfers, 321–2
- and speed, 111, 115, 116
- support for functional stall, 54

interrupt transfers

- about, 41, 42, 46, 72
- availability, 72
- vs. bulk transfers, 41, 46
- vs. control transfers, 41, 46
- data size, 73
- data toggles in, 58
- error handling, 75–6
- in HID transfers, 321, 322, 322–3
- vs. isochronous transfers, 41, 46
- maximum data-transfer rate per endpoint, 62
- receiving HID reports via, 343, 344, 345
- sending HID reports via, 340, 341, 342
- speed, 73–5
- and stream pipes, 43
- structure, 70, 72–3
- transactions, 36, 46

IN transactions, 38–9, 39, 45, 53

I/O boards, 152–3, 153

I/O interfaces, 147–8

I/O Request Packets (IRPs), 475, ch8

IrDA Bridge class, 179, 206–8, 534

IrDA interface *vs.* USB, 3

isochronous endpoints, 81

isochronous transfers

- about, 41, 42, 46, 76
- availability, 76
- vs. bulk transfers, 41, 46
- vs. control transfers, 41, 46
- data size, 79
- data toggles in, 58–9
- error handling, 80
- vs. interrupt transfers, 41, 46

- maximum data-transfer rate per endpoint, 62
- speed, 79–80
- and stream pipes, 43
- structure, 76–9, 77
- transactions, 36, 46

isolation, galvanic, 530

J

Jungo's WinDriver USB Device toolkit, 248

K

kernel mode, 235, 237, 249

Keyboard device setup class, 251

Keyspan hub/server, 525, 534

L

latencies

- minimizing, 427–8
- under Windows, 83–4

layered driver model

- about, 234, 236, 238
- client drivers, 238–40, 239, 240, 245
- host-controller drivers, 239, 241–2
- major components, 236
- USB drivers, 239, 240–1
- WDM driver overview, 237–8

LeCroy Corporation, 477

legacy hardware, 11–12

Linux, support for USB, 7–8

Local Item type

- defined, 370
- Delimiter item, 371, 372
- Designator Index item, 371, 372
- Designator Maximum item, 371, 372
- Designator Minimum item, 371, 372
- Reserved item, 371
- String Index item, 371, 372
- String Maximum item, 371, 372
- String Minimum item, 371, 372
- Usage item, 370–1, 371
- Usage Maximum item, 371, 372
- Usage Minimum item, 371, 372

logging events during device installation, 277

Logical Maximum item, 362, 364–5
 Logical Minimum item, 362, 364–5
 logo, USB, 477, 484, 485. *See also* icon, USB
 Long items, defined, 354–5
 low speed, defined, 5
 low-speed bus states, 490–2
 low-speed keep alive signal, 48–9
 low-speed transceivers, 508–12
 Lucent, and USB 2.0 specification, 14

M

Macintosh, support for USB, 7–8

Main item type

- Collection items, 360–1, 361
- defined, 355
- End Collection item, 360–1, 361
- Feature items, 356, 357, 358–60
- Input items, 356, 357, 358–60
- Output items, 356, 357, 358–60

managed code, 282–4

marshaling, 283

mass-storage class, 179, 208–13, 251
 INF file, 276

memory

- for data, 147
- electrically erasable PROM, 146–7
- erasable programmable ROM, 146
- flash, 146
- one-time programmable ROM, 146
- random-access, 147
- read-only, 146

message pipes, 42

Micrel, Inc., 466

Microchip Technology

- code for HID data transfer example, 338–49
- controller chips, 155
- PIC16C745 microcontroller, 155, 157
- PIC16C765 microcontroller, 155, 157
- PIC16F877 microcontroller in FT232BM device controller example, 408, 409, 410, 411–13, 414

- PIC16F877 microcontroller in FT245BM device controller example, 414, 415, 416, 417, 418–21, 421

- PIC18F4550 microcontroller, 155, 156–7, 338

microcontrollers, defined, 9

microframes

- in bulk transfers, 71
- in control transfers, 66, 67
- defined, 24, 35
- and frame numbers, 48, 501
- illustrated, 35
- in interrupt transfers, 73–5
- in isochronous transfers, 76–80
- role in USB transfer, 35–6
- in split transactions, 443
- and timing accuracy, 499
- in transactions, 48, 50, 59, 66

Microsoft Corporation. *See also* Windows operating systems

- and OHCI standard, 241

- testing opportunities, 477

- and USB 2.0 specification, 14

- WDM drivers, 8, 237–47

- Windows Hardware Quality Labs testing, 484–7

- and Wireless USB Promoter Group, 533

Microsoft Intermediate Language (MSIL), 282

Microsoft .NET, 243–4, 281–4. *See also* Visual Basic .NET; Visual C++ .NET

Microsoft OS descriptor, 114

Microsoft Windows logo, 477, 485–6

Microwire interface *vs.* USB, 3

MIDI interface *vs.* USB, 3

MIDIStreaming interface, 185–8, 186

miniport drivers, 240

Modem device setup class, 251

Mouse device setup class, 251

MS-DOS, 11

MSIL (Microsoft Intermediate Language), 282

multi-tasking, Windows, 83–4

N

- NAK (negative acknowledge) handshake code, 48, 52, 53, 55, 56, 56
- National Semiconductor
 - and OHCI standard, 241
 - USBN9603 controller chip, 165–7, 166
- NEC
 - and USB 2.0 specification, 14
 - and Wireless USB Promoter Group, 533
- .NET platform. *See* Microsoft .NET; Visual Basic .NET; Visual C++ .NET
- noise, sources of, 526
- Non-Return to Zero Inverted (NRZI) with bit stuffing encoding, 494–5, 495, 496
- NRZI (Non-Return to Zero Inverted (NRZI) with bit stuffing) encoding, 494–5, 495, 496
- NYET (not yet) handshake code, 48, 52, 53, 54–5, 56

O

- OHCI (Open Host Controller Interface), 239, 239, 241
- OnDeviceChange method, 312
- one-time programmable (OTP) ROM memory, 146
- On-The-Go (OTG) devices
 - about, 536
 - cables, 542, 543, 544
 - capabilities and limits, 536–8, 537
 - connectors, 542, 543, 544
 - controller chips, 545–9
 - Cypress CY7C67200 EZ-OTG controller chip, 548–9
 - and peer-to-peer communication, 10, 16, 229–30, 230
 - Philips ISP1261 bridge controller, 549
 - Philips ISP1362 controller chip, 546–7
 - remote wakeup support, 542
 - requirements, 538–44
 - targeted peripheral list, 544
 - TransDimension TD242LP, 547–8
- Open Host Controller Interface (OHCI), 239, 239, 241

- operating systems, support for USB, 7–8, 14, 15. *See also* Windows operating systems
- OTG. *See* On-The-Go (OTG) devices
- OTG descriptor, 95, 113, 545, 546
- other_speed_configuration descriptor, 94, 95, 103, 104
- OTP (one-time programmable) ROM memory, 146
- Output items, 356, 357, 358–60
- Output reports
 - Main item types in, 356, 357, 358–60
 - sending to devices, 389–92
 - transfer types, 389, 389
- OUT transactions, 38–9, 39, 45, 53
- over-current protection, 462–3

P

- Packet IDs (PIDs)
 - about, 45–6
 - data-toggle value, 57–9
 - defined, 500, 500
 - ERR, 48, 49
 - error checking, 57
 - PING, 48, 49
 - PRE, 48, 49
 - special, 48, 49
 - SPLIT, 48, 49
 - table, 47–8
 - in transaction data phase, 46, 47
 - in transaction handshake phase, 47, 48
 - in transaction token phase, 46, 47, 48–9
- packets
 - data, 44, 44, 45, 46, 47, 49–50, 52, 53
 - fields in, 499–501, 500
 - handshake, 44, 45, 46, 47, 48, 50, 52–3, 53
 - inter-packet delay, 501
 - sequence, 49–50
 - short, 50
 - time issues, 50
 - token, 44, 45, 46, 47, 48, 49
 - as transaction element, 45–51
- padding descriptors, 373
- parallel ports
 - converters for, 229

vs. USB, 3

PCs. *See also* host computers

- legacy hardware, 11–12
- power issues, 460, 460, 462, 463, 467–9, 468
- support for USB, 7–8
- using to emulate devices, 153–4

peer-to-peer communication, 10, 16, 229–30

peripherals. *See* devices

Philips Semiconductor

- ISP1181B controller, 166, 167–8
- ISP1261 bridge controller, 549
- ISP1362 controller, 546–7
- ISP1581 controller, 166, 168–9
- PDIUSB12 controller, 166
- and USB 2.0 specification, 14
- and Wireless USB Promoter Group, 533

physical descriptors, 373

Physical Maximum item, 362, 365, 366

Physical Minimum item, 362, 365, 366

PIC16C745 microcontroller, 155, 157

PIC16C765 microcontroller, 155, 157

PIC16F877 microcontroller, 408, 409, 410, 411–13, 414, 415, 416, 417, 418–21, 421

PIC18F4550 microcontroller, 155, 156–7, 338

PID field. *See* Packet IDs (PIDs)

PING PID, 48, 49, 53, 54

pipes

- about, 40
- Default Control Pipe, 40, 62, 66
- defined, 40
- message, 42
- stream, 43
- and transfer types, 40–2

plugs. *See* cables; connectors

PLX Technology

- NET2272 controller, 166, 169–70
- NET2272 PCI-RDK development kit, 153–4

PNF files, 278

Pop item, 362, 370

port drivers, 213, 216, 240. *See also* host-controller drivers

port indicators, 453, 454

ports

- about, 21–2
- parallel, 3, 229
- USB, 22, 27
- user status indicators, 453, 454

power consumption, USB, 6, 455–69

power supplies, 4–5, 25, 26, 455–6, 457, 458. *See also* bus power; self-powered devices

power switching, 463

PRE PID, 48, 49

printer class, 179, 213–17, 251, 276

Product ID

- and composite devices, 274–5
- creating device ID from, 272
- customizing, 429, 486
- default, 424–6, 432
- and FTDI chip controllers, 422, 424
- in INF files, 269, 278, 486
- obtaining, 13
- reading, 380–4
- in Windows registry, 258, 258–9, 261

program memory, in USB devices, 145–7

programming languages, as necessary tool for USB device development, 30. *See also* Visual Basic .NET; Visual C++ .NET

Prolific Technology, 155, 230

Properties pages, Windows Device Manager, 257, 279

protocol analyzers

- about, 471–2
- Ellisys USB Explorer, 473, 474, 474–5, 475
- hardware, 472, 472–5, 474, 475
- as recommended tool for USB device development, 30
- software, 475–7, 476

pull-down resistors, 446, 509, 511, 512

pull-up resistors, 446, 446, 465, 465, 509, 510, 511, 512, 513

Push item, 362, 369

R

RAM (random-access memory), 147

random-access memory (RAM), 147

Index

- ReadFile API function, 376, 379, 392–401
- read-only memory (ROM), 146
- receptacles. *See* cables; connectors
- RegisterDeviceNotification API function
 - about, 307
 - defined, 308
 - in Visual Basic code, 309–11
 - in Visual C++ code, 308–9
- registers, in USB controllers, 144–5
- registry, Windows
 - about, 257–8
 - class keys, 259–60, 260
 - driver keys, 260–2, 261, 279
 - hardware keys, 258, 258–9, 261, 279–80
 - service keys, 262, 263, 263, 280
 - viewing contents, 257
- reliability, USB, 5
- Remote NDIS devices, 195
- remote wakeup feature
 - enabling/disabling, 468
 - for OTG devices, 542
 - power issues, 464, 465
 - workarounds for older versions of Windows, 468–9
- repeaters, hub, 435–8, 436
- Report Count item, 362, 369
- report descriptors, HID, 328–30, 329
- Report ID item, 361–3, 362
- reports, HID
 - about, 322, 323, 351–2
 - API functions for managing communications, 379, 381
 - API functions for providing and using report data, 378–9, 380
 - API functions for sending and receiving, 376–8, 379
 - control values, 354
 - converting raw data, 366–9, 367
 - converting units, 365–6
 - defined, 351
 - describing data size and format, 369
 - describing use, 363–5
 - Global item type, 361–70
 - identifying, 361–3
 - item types, 354–5
 - Local Item type, 370–3
 - Main item type, 355–61
 - physical descriptors, 373
 - reading Input reports from devices, 392–401
 - receiving via control transfers, 347–8, 349
 - receiving via interrupt transfers, 343, 344, 345
 - saving and restoring Global items, 369–70
 - sending via control transfers, 345, 346, 347, 348
 - sending via interrupt transfers, 340, 341, 342
 - structure, 351–5
 - transfer types, 389, 389
 - writing Output reports to devices, 389–92
- Report Size item, 362, 369
- requests
 - about, 127
 - class-specific. *See* class-specific requests
 - Clear_Feature request, 128, 130
 - Get_Configuration request, 128, 135
 - Get_Descriptor request, 94, 128, 133, 452
 - Get_Interface request, 128, 137
 - Get_Status request, 128, 129
 - Set_Address request, 128, 132
 - Set_Configuration request, 128, 136
 - Set_Descriptor request, 128, 134
 - Set_Feature request, 128, 131
 - Set_Interface request, 128, 138
 - Synch_Frame request, 128, 139
 - vendor-specific, 140
- Reserved item, 362, 371
- Reset bus state, 492
- Reset_TT request, 451, 452
- resistors, pull-up *vs.* pull-down, 446, 509, 511, 512
- Resume bus state, 491
- Resume state, 466
- ROM (read-only memory), 146
- Root 2 USB Test Host, 477
- root hub, 17, 18, 22, 23, 86, 87, 254, 434, 435, 461
- RPM Systems, 477

- RS-232 interface
 - converters for, 11, 227–9, 228
 - single-ended lines, 527
 - vs. USB, 3, 10, 22
 - RS-485 interface
 - converters for, 11
 - vs. USB, 3, 10
- S**
- Samsung Electronics, and Wireless USB Promoter Group, 533
 - SBAE-30 Analyzer/Exerciser, 477
 - self-powered devices, 456, 457, 458–9, 462
 - serial interface engines (SIEs)
 - in Microchip PIC18F4550, 157
 - in PC-to-PC communication, 230, 230
 - in USB controllers, 143
 - service keys, Windows registry, 262, 263, 263, 280
 - Session Request Protocol (SRP), 541–2, 545
 - Set_Address request, 128, 132, 452
 - Set_Configuration request, 128, 136, 452
 - Set_Descriptor request, 128, 134
 - Set_Feature request, 128, 131, 452, 545
 - Set_Hub_Descriptor request, 451
 - Set_Hub_Feature request, 451, 452, 453
 - Set_Idle request, 331, 336
 - Set_Interface request, 128, 138, 452
 - Set_Port_Feature request, 451, 452, 454
 - Set_Protocol request, 331, 337
 - Set_Report request, 331, 335
 - SetupDi_ API functions, defined, 291, 292
 - SetupDiDestroyDeviceInfoList API function
 - defined, 292
 - in Visual Basic code, 303
 - in Visual C++ code, 302
 - SetupDiEnumDeviceInterfaces API function
 - about, 295, 297–8
 - defined, 292
 - in Visual Basic code, 296–7
 - in Visual C++ code, 295–6
 - SetupDiGetClassDevs API function
 - about, 294–5
 - defined, 292
 - in Visual Basic code, 294
 - in Visual C++ code, 293
 - SetupDiGetDeviceInterfaceDetail API function
 - about, 298
 - defined, 292
 - extracting device path names, 301–2
 - requesting structures containing device path names, 298–301
 - in Visual Basic code, 300–1
 - in Visual C++ code, 298–9
 - Setup stage
 - about, 46, 118
 - in control transfer structure, 62, 63, 64, 65
 - data packets, 118–20
 - handshake packets, 120
 - token packets, 118
 - Setup transactions, 39, 39, 42, 45, 53, 63, 64, 65, 118
 - shielding, 528–9
 - Short items, defined, 354–5
 - short packets, 50
 - SIEs (serial interface engines)
 - in Microchip PIC18F4550, 157
 - in PC-to-PC communication, 230, 230
 - in USB controllers, 143
 - signal quality
 - edge rate factor, 529
 - noise sources, 526
 - role of balanced lines, 527
 - role of galvanic isolation, 529–30
 - role of shielding, 528–9
 - role of twisted pairs, 527–8, 528
 - signal voltages, 517–18
 - signed drivers, 274
 - Silicon Laboratories C8051F32x controllers, 151, 155
 - Single-Ended-One (SE1) bus state, 490
 - Single-Ended-Zero (SE0) bus state, 490, 498
 - smart-card readers. *See* CCIDs (Chip Card Interface Devices)
 - Smart Card Readers device setup class, 251
 - smart cards, 189. *See also* chip/smart card interface class
 - SOF (Start-of-Frame) packets, 48, 447, 464, 498
 - soft-start capability, 466
 - software. *See* firmware, defined

Index

- software protocol analyzers, 475–7, 476
 - Sourcequest, Inc., 476
 - SourceUSB software analyzer, 476, 476–7
 - speed. *See* bus speed
 - SPI interface *vs.* USB, 3
 - SPLIT PID, 48, 49
 - split transactions, 50–1, 439–44, 441, 442
 - SRP (Session Request Protocol), 541–2, 545
 - STALL handshake code, 48, 52, 53, 53–4, 55, 56, 56
 - Standard Microsystems Corporation (SM-SC), 155
 - Start bit, 496
 - Start-of-Frame (SOF) packets
 - about, 48
 - End-of-High-speed-Packet in, 498
 - error-checking bits, 56
 - and Resume state, 466
 - and Suspend state, 447, 464
 - Start-of-High-speed-Packet (HSSOP) bus state, 494
 - Start-of-Packet (SOP) bus state, 491
 - star topology, 18, 18–19
 - status packets, defined, 48. *See also* handshake packets
 - status phase, defined, 48. *See also* handshake packets
 - Status stage
 - about, 46, 122
 - in control transfer structure, 56, 62, 63, 64, 65
 - data packets, 56, 123
 - handshake packets, 56, 123–4
 - token packets, 122–3
 - still-image class, 179, 217–20, 251, 276
 - STMicroelectronics, 155
 - Stop bit, 496
 - Stop_TT request, 451, 452
 - stream pipes, 43
 - string descriptor, 94, 95, 112–13, 113
 - String Index item, 371, 372
 - String Maximum item, 371, 372
 - String Minimum item, 371, 372
 - strings, API functions for receiving, 376, 378
 - structures, passing in Visual Basic .NET, 290
 - Suspend state, 447, 457, 458, 464–6, 465, 516–17
 - SYNC field, 497, 499, 500
 - Synch_Frame request, 128, 139, 452
 - synchronization, 496–8
 - system registry. *See* registry, Windows
- ## T
- Tape Drives device setup class, 251
 - targeted peripheral list, 544
 - Terminal descriptor, 183, 183–4
 - Tetradyne Software, 231
 - test-and-measurement class (USBTMC), 179, 220–1
 - test modes
 - about, 502
 - entering, 502
 - exiting, 502
 - Test_Force Enable, 503
 - Test_K, 503
 - Test_Packet, 503
 - Test_SEO_NAK, 502–3
 - Texas Instruments, 155
 - Thesycon Systemsoftware & Consulting GmbH, 231
 - TIA/EIA-485. *See* RS-485
 - tiered-star topology, 18, 18–19
 - token packets, 44, 45, 46, 47, 48, 49, 118, 121, 122–3
 - token phase, transactions. *See* token packets
 - topology, USB, 18–19
 - traffic generators, 477
 - transactions
 - about, 37, 44–5
 - data phase, 45, 46, 47, 49–50, 52, 53
 - defined, 44
 - handshake phase, 45, 46, 47, 48, 50, 52, 53
 - IN, OUT, and Setup types, 39–40, 45
 - multiple, 50
 - packets in, 45–6, 46
 - phases, 45–51
 - role in USB transfers, 36, 44–5
 - split, 50–1, 439–44, 441, 442
 - token phase, 45, 46, 47, 48, 49
 - as transfer element, 36, 37, 44–5
 - transaction translators, 438–44, 439

- transceivers
 - full-speed, 508–12, 510
 - high-speed, 512–17, 513, 514
 - low-speed, 508–12, 510
 - low-speed *vs.* full-speed, 508
 - in USB controllers, 143
 - TransDimension TD242LP, 547–8
 - transfers
 - application communications, 34
 - bulk type. *See* bulk transfers
 - communications flow, 243–6
 - control type. *See* control transfers
 - elements, 37–51
 - ensuring success, 51–9
 - enumeration communications, 34
 - examples, 243–7
 - illustrated, 44
 - initiating, 43–4
 - interrupt type. *See* interrupt transfers
 - isochronous type. *See* isochronous transfers
 - role of device drivers, 235–7
 - signal quality issues, 525–30
 - speed-limiting factors, 80–4
 - transactions, 36, 37, 39–40, 44–5, 46
 - type overview, 40–2, 46
 - trusted drivers, 274, 275, 276
 - twisted-pair cables, 527–8, 528
- ## U
- UHCI (Universal Host Controller Interface), 239, 239, 241
 - Unit descriptor, 183, 184, 187
 - Unit Exponent item, 365, 366
 - Unit item, 362, 366, 367
 - Universal Host Controller Interface (UHCI), 239, 239, 241
 - Universal Serial Bus. *See* USB (Universal Serial Bus)
 - Universal Serial Bus Specification*, 33
 - unmanaged code, 282–4
 - UnregisterDeviceNotification API function
 - about, 317
 - defined, 308
 - in Visual Basic code, 318
 - in Visual C++ code, 318
 - untrusted drivers, 276
 - Usage item, 370–1, 371
 - Usage Maximum item, 371, 372
 - Usage Minimum item, 371, 372
 - Usage Page item, 362, 363–4
 - USB 1.x specification
 - about, 14–15
 - cable requirements, 524–5
 - hub descriptors, 448
 - low and full speeds, 5
 - making descriptors 2.0 compliant, 114–15
 - USB 2.0 specification
 - about, 15–16
 - cable requirements, 524–5
 - hub descriptors, 448
 - making 1.x descriptors compliant, 114–15
 - USB Command Verifier (USBCV) utility, 479, 479–81, 480
 - USB controllers. *See* device controllers
 - USBCV (USB Command Verifier) utility, 479, 479–81, 480
 - USB devices. *See* devices
 - USB device setup class, 251
 - USB drivers. *See* device drivers
 - USB icon, 522, 523. *See also* USB logo
 - USB-IF (USB Implementers Forum)
 - about, 9
 - compliance program, 478–84, 479, 480, 483
 - Device Working Groups, 178
 - HID Descriptor Tool, 352, 353
 - testing opportunities, 477
 - USBIO Development Kit, 231
 - USB logo, 477, 484, 485. *See also* USB icon
 - USB On-The-Go (OTG). *See* On-The-Go (OTG) devices
 - USB ports, 22, 27
 - USB Request Blocks (URBs), 245, 475
 - USBTMC. *See* test-and-measurement class (USBTMC)
 - USB transfers. *See* transfers
 - USB (Universal Serial Bus)
 - about, 1–2
 - benefits for developers, 6–9
 - benefits for users, 2–6

Index

- ease of use, 2–5
- vs. Ethernet, 3, 10, 17
- vs. FireWire 800, 3
- vs. GPIB, 3
- history, 13–17
- vs. I²C, 3
- vs. IEEE-488, 3
- vs. IEEE-1394, 3, 10, 16–17
- vs. IrDA, 3
- limitations, 10–12
- vs. Microwire, 3
- vs. MIDI, 3
- operating system support, 7–8, 14, 15, 180
- vs. other computer interfaces, 3
- vs. parallel printer port, 3
- physical components, 17–22
- reliability, 5
- vs. RS-232, 3, 10, 22
- vs. RS-485, 3, 10
- speed issue, 5, 15–16
- vs. SPI, 3
- topology, 18–19
- transmission path, 35–6
- Version 1.x specification, 14–15
- Version 2.0 specification, 15–16
- user mode, 235–6

V

- VCP (Virtual COM Port) driver, 174, 198, 227, 421, 426
- Vendor ID
 - and composite devices, 274–5
 - creating device ID from, 272
 - customizing, 429, 486
 - default, 424–6, 432
 - and FTDI chip controllers, 422, 424
 - in INF files, 269, 278, 486
 - obtaining, 13
 - reading, 380–4
 - validity issue, 278
 - in Windows registry, 258, 258–9, 261
- vendor-specific requests, 140
- video class
 - about, 222, 222–3
 - chips, 225
 - descriptors, 223–5
 - documentation, 222
 - where declared, 179
- VideoControl interface, 223–4, 224
- VideoStreaming interface, 224–5, 225
- Virtual COM Port (VCP) driver, 174, 198, 227, 421, 426
- Visual Basic .NET
 - ByRef and ByVal in, 289
 - calling API functions in, 282–3
 - capturing device change messages, 311
 - closing communications, 303
 - closing device communication, 406
 - closing handles, 306
 - data types, 288
 - declarations for API functions, 287–8
 - DLL names in code, 288
 - extracting device path names, 302
 - getting pointer to buffer with device capabilities, 385
 - HidD_GetAttributes API function in, 382–3
 - HidD_GetFeature API function in, 405
 - HidD_GetPreparedData API function in, 385
 - HidD_SetFeature API function in, 403
 - HidP_GetCaps API function in, 387
 - identifying device interfaces, 296–7
 - obtaining device interface GUID, 293
 - passing structures, 290
 - reading device change messages, 313–14
 - reading device path name in message, 316–17
 - reading Input reports from devices, 396–9
 - reading Vendor ID and Product ID, 382–3
 - registering for device notifications, 309–11
 - requesting communications handles, 304–5
 - requesting pointer to device information set, 293
 - requesting structures containing device path names, 300–1
 - stopping device notifications, 318

- using Windows API functions in applications, 286–91
- vs.* Visual C++, 286
- writing Output reports to devices, 391
- Visual C++ .NET
 - calling API functions in, 282–3
 - capturing device change messages, 311
 - closing communications, 302
 - closing device communication, 406
 - closing handles, 306
 - extracting device path names, 302
 - getting device capabilities, 386, 387
 - getting pointer to buffer with device capabilities, 384
 - HidD_GetAttributes API function in, 381–2
 - HidD_GetFeature API function in, 404
 - HidD_GetPreparedData API function in, 384
 - HidD_SetFeature API function in, 402
 - HidP_GetCaps API function in, 386
 - identifying device interfaces, 295–6
 - managed *vs.* unmanaged code in, 283
 - obtaining device interface GUID, 292
 - reading device change messages, 312–13
 - reading device path name in message, 315
 - reading Input reports from devices, 394–6
 - reading Vendor ID and Product ID, 381–2
 - registering for device notifications, 308–9
 - requesting communications handles, 303–4
 - requesting pointer to device information set, 293
 - requesting structures containing device path names, 298–9
 - stopping device notifications, 318
 - using Windows API functions in applications, 284–6
 - writing Feature reports to devices, 402, 403
 - writing Output reports to devices, 390
- Visual C# .NET, calling API functions in, 282–4

W

- waivers, USB-IF, 484
- WDM (Windows Driver Model) drivers, 8, 237–47
- WHQL (Windows Hardware Quality Labs), 484–7
- WindowProc function, 307, 311
- Windows 98
 - support for USB, 7–8, 14, 180
 - troubleshooting remote wakeup, 469
- Windows 2000, support for USB, 7–8, 180
- Windows API functions
 - about, 243–5, 281–4
 - documentation, 284
 - HID, 375–9
 - in Visual Basic .NET applications, 286–91
 - in Visual C++ applications, 284–6
- Windows Device Manager, 253–7, 254, 255, 256
 - viewing device information in, 254–7
 - viewing Properties pages, 257, 279
- Windows Driver Development Kit (DDK), 30, 237
- Windows Driver Model. *See* WDM (Windows Driver Model) drivers
- Windows Hardware Quality Labs (WHQL), 274, 484–7
- Windows logo, 477, 485–6
- Windows Me
 - support for USB, 7–8, 180
 - troubleshooting remote wakeup, 469
- Windows operating systems
 - power issues, 460, 460, 462, 463, 467–9, 468
 - searching for drivers, 279–80
 - support for USB, 7–8, 180
 - system registry, 257–62
- Windows registry
 - about, 257–8
 - class keys, 259–60, 260
 - driver keys, 260–2, 261, 279
 - hardware keys, 258, 258–9, 261, 279–80
 - service keys, 262, 263, 263, 280
- Windows Server 2003, support for USB, 7–8
- Windows XP, support for USB, 7–8, 180

Index

winDriver USB Device toolkit, 248
wireless bridges, 534
wireless connections, 530–1, 532, 533
Wireless USB Promotor Group, 533
Wireless USB specification, 16
WirelessUSB (Cypress), 530–1, 532, 533

WM_DEVICECHANGE message, 307,
311
WriteFile API function, 376, 379, 389–92

Z

ZigBee, 534