# 20

# Dual-role Devices with USB On-The-Go

A USB host in a desktop system has many responsibilities, including supporting three bus speeds, managing communications with multiple devices, and providing up to 500 milliamperes to every device connected to the root hub. PCs and other desktop computers typically have the resources to implement a full USB host. But many smaller systems could benefit if they could function as hosts as well. For example, a camera could connect directly to a USB printer. A data-acquisition device could store its data in a USB drive. A PDA could interface to a USB keyboard and mouse. Two drives could exchange files.

An embedded system can incorporate a limited-capability host that supports communications with just one or a few devices. But for small systems, implementing even a limited-capability USB host can be challenging. The CPU may have limited resources, and battery-powered systems may be unable to provide the bus power that the host must make available. And a

USB device that also functions as a host requires two connectors: a Series-A receptacle for the host and a Series-B or mini-B receptacle for the device.

The On-The-Go (OTG) Supplement to the USB 2.0 Specification offers a an alternative for small devices that also want to function as hosts. The supplement defines a way for a USB device to function as a host with limited capabilities that are within the reach of many simpler devices. Version 1.0 of the On-The-Go supplement was released in 2001.

# Device and Host in One

An OTG device is a dual-role device that can function both as a limited-capability host and as a USB peripheral. When functioning as a host, the OTG device can communicate with the devices in its targeted peripheral list. The list can be as limited as a single device or as extensive as a series of device types (keyboard, mouse, mass storage). The targeted peripherals can be any combination of other OTG devices and peripheral-only devices.

## Capabilities and Limits

Table 20-1 compares the requirements of an On-The-Go device functioning as a host and a conventional, non-On-The-Go host. An OTG host doesn't have to support external hubs, multiple devices attached at the same time, or high and low speeds. The USB hosts in desktop systems support all three speeds and have multiple ports. The USB 2.0 specification doesn't forbid hosts with more limited capabilities, however.

Because On-The-Go communications often involve battery-powered devices, conserving power is important. For this reason, an OTG device functioning as a host is allowed to turn off the VBUS voltage when the bus is unused. Communications occur in sessions, with a session beginning when VBUS is above the session valid-threshold voltage and ending when VBUS falls below this voltage. The Session Request Protocol (SRP) enables a device to request a session even if VBUS isn't present.

The On-The-Go supplement defines new connector types in addition to the Series A, Series B, and mini-B plugs and receptacles defined in the USB 2.0

Table 20-1: Compared to a non-OTG host, an OTG device functioning as a host doesn't have to supply as much power and can use a single connector for host and peripheral functions.

| Capability or Feature | Non-OTG Host | OTG Device Functioning as a Host |
|---|---|---|
| Communicate at high speed | Hosts in desktop systems support all three speeds. Hosts in embedded systems can support one or more speeds. | optional |
| Communicate at full speed | | yes |
| Communicate at low speed | | optional in host mode; not allowed in device mode |
| Allow external hubs | yes | optional |
| Provide targeted peripheral list | no | yes |
| Function as a peripheral | no | yes |
| Support Session Request Protocol | optional | yes |
| Support Host Negotiation Protocol | no | yes |
| Minimum available bus current per port | 500 mA (100 mA if battery-powered) | 8 mA |
| OK to turn off VBUS when unneeded? | no | yes |
| Connector | 1 or more standard A | 1 mini-AB |

specification. The mini-A plug is a smaller alternative to the Series-A plug. The mini-AB receptacle can accept either a mini-A plug or a mini-B plug. Figure 20-1 shows mini-AB and mini-B receptacles. Every OTG device

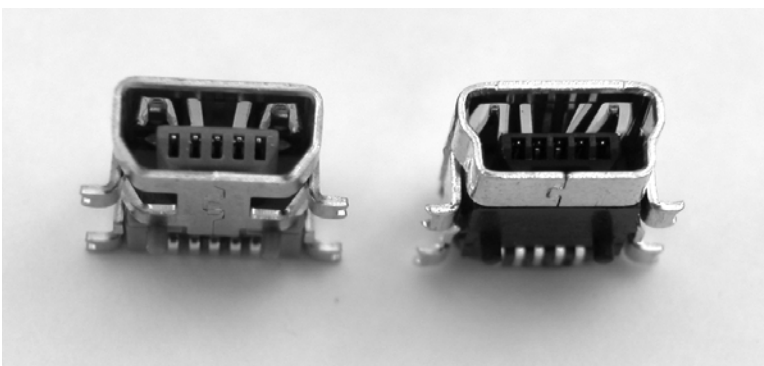

Figure 20-1: A mini-AB receptacle (left) accepts a mini-A or mini-B plug. A mini-B receptacle (right) accepts only a mini-B plug.

must have a mini-AB receptacle. The only approved use for the mini-A receptacle is in an adapter that converts a mini-A plug to a Series-A plug.

On every approved USB cable, one end has a Series A or mini-A plug and the other end has a Series-B plug, a mini-B plug, a vendor-specific connector, or a permanent attachment to a device.

Every On-The-Go connection is between an A-device and a B-device. The A-device is defined by the type of plug inserted in the device's USB receptacle. The device with a Series A or mini-A plug is the A-device, and the device at the other end of the cable is the B-device. The A-device initially functions as the host, and the B-device initially functions as the peripheral. Two connected OTG devices can use a protocol to swap functions when needed, as described below. The A-device always provides the VBUS voltage and current, even when functioning as a peripheral.

## Requirements for an OTG Device

An OTG device must provide all of the following:

- The ability to function as a full-speed peripheral. Support for high speed is optional. The peripheral must not use low speed.
- The ability to function as a host that can communicate with one or more full-speed devices. Support for low- and high-speed communications is optional. Support for hubs is optional.
- Support for the Host Negotiation Protocol, which enables two OTG devices to swap roles. (The host becomes the peripheral and the peripheral becomes the host.)
- The ability to initiate and respond to the Session Request Protocol, which enables a device to request communications with the host even if VBUS isn't present.
- Support for remote wakeup.
- One and only one Mini-AB receptacle, which can accept either a Mini-A plug or a Mini-B plug.
- The ability to provide at least 8 milliamperes of bus current when functioning as the A-device.

- A display or other way to communicate messages to users.
- A targeted peripheral list that names the devices the host can communicate with.

On-The-Go adds complexity by requiring hosts to support HNP and SRP and to be able to function as peripherals. On the other hand, On-The-Go reduces complexity by using a single connector for the host and device roles and by not requiring the host to supply large bus currents or support external hubs.

The following paragraphs describe the requirements for OTG devices in more detail.

### Full-speed Device Capability

Any device that implements On-The-Go's limited-capability host must also be able to function as a USB peripheral. OTG host-only products aren't allowed. When functioning as a peripheral, an OTG device may support high speed and must not communicate at low speed.

### Full-speed Host Capability

An OTG device functioning as a host must be able to communicate with one or more devices. The host must support full speed and may support low speed and/or high speed. The host does not have to support communications via hubs.

### The Host Negotiation Protocol

The Host Negotiation Protocol (HNP) enables the B-device to request to function as a host. When connecting two OTG devices to each other, users don't have to worry about which end of the cable goes where. When necessary, the devices use HNP to swap roles.

When two OTG devices are connected to each other, the A-device enumerates the B-device in the same way that a standard USB host enumerates its devices. During enumeration, the A-device retrieves the B-device's OTG descriptor, which indicates whether the B-device supports HNP. If the B-device supports HNP, the A-device can send a Set_Feature request with a request code of hnp_enable. This request informs the B-device that it can use HNP to request to function as the host when the bus is suspended.

At any time after enumerating, if the A-device has no communications for the B-device, the A-device can suspend the bus. A B-device that supports HNP may then request to communicate. The B-device can use HNP in response to user input such as pressing a button, or firmware can initiate HNP without user intervention.

Standard hubs don't recognize HNP signaling. If there is a hub between the B-device and the A-device, the A-device must not send the hnp_enable request and the B-device can't use HNP.

This is the protocol the B-device uses to request to operate as the host:

1. The A-device suspends the bus.

2. If the devices were communicating at full speed, the B-device disconnects from the bus by switching off its pull-up resistor on D+. If the devices were communicating at high speed, the B-device switches on its pull-up resistor on D+ for 1 to 147 milliseconds, then switches the resistor off. The bus is then in the SE0 state.

3. The A-device detects the SE0 state and connects to the bus as a device by switching on its pull-up resistor on D+. The bus is in the J state.

4. The B-device detects the J state and resets the bus.

5. The B-device enumerates the A-device and can then perform other communications with the device.

When the B-device is finished communicating, it returns to its role as a peripheral using the following protocol:

1. The B-device suspends the bus and may switch on its pull-up resistor.

2. The A-device detects the suspended bus and switches off its pull-up resistor or removes VBUS to end the session.

3. If the B-device didn't switch on its pull-up resistor in Step 1, the B-device switches on its pull-up resistor to connect as a peripheral. The bus is in the J state.

4. If VBUS is present, the A-device detects the J state and resets the bus. The A-device can then enumerate and communicate with the B-device, suspend the bus, or end the session by removing VBUS.

The A-device and B-device must also control their pull-down resistors on D+ and D-. When idle or functioning as a host, an OTG device should switch on its pull-down resistors. When functioning as a peripheral, an OTG device should switch off its pull-down resistor on D+ only.

## The Session Request Protocol

If the A-device has turned off the VBUS voltage, a B-device can use the Session Request Protocol (SRP) to request the host to restore VBUS and begin a new session. There are two SRP methods: data-line pulsing and VBUS pulsing. The B-device must try data-line pulsing first, followed by VBUS pulsing. An A-device that supports SRP must respond to one of the methods. OTG devices must support SRP both as an A-device and as a B-device. Other hosts and devices may support SRP, but aren't required to.

In data-line pulsing, the device turns on its pull-up resistor (on D+ or D-, depending on device speed) for 5 to 10 milliseconds.

In VBUS pulsing, the device must drive the VBUS line long enough for the host to detect the VBUS voltage but not long enough to damage a non-OTG host that isn't designed to withstand a voltage applied to VBUS. To meet this requirement, the B-device should drive VBUS until the voltage is greater than 2.1V if connected to a OTG device and less than 2.0V if connected to a non-OTG host. The device can do so because of the difference in capacitance at the two host types. On a non-OTG host, the VBUS capacitance is 96 microfarads or more, while on a OTG device, the VBUS capacitance is 6.5 microfarads or less.

To ensure that the VBUS current doesn't exceed 8 milliamperes even if the A-device drives VBUS while the B-device is pulsing VBUS, the B-device can use a voltage source greater than 3V with an output impedance greater than 281 ohms.

Within 5 seconds of detecting data-line pulsing or VBUS pulsing, the A-device must turn on VBUS and reset the bus.

Standard hubs don't recognize SRP signaling, so if there is a hub between the B-device and the A-device, the B-device can't use SRP. Any non-OTG USB peripheral also has the option to support SRP.

### Support for Remote Wakeup

When VBUS is present and the bus is suspended, an OTG device can use remote wakeup to request communications from an OTG device or other USB host.

### Cables and Connectors

If you see a mini-AB receptacle, you know you have an OTG device. Every OTG device must have one and only one Mini-AB receptacle, and any device with a mini-AB connector must function as a OTG device. The mini-AB receptacle can accept either a Mini-A plug or a Mini-B plug. A host or upstream hub connects to the mini-AB receptacle with a mini-B plug, a peripheral connects with a Mini-A plug, and an OTG device can connect using either plug type.

Figure 20-2 shows the cabling options. A cable that connects two OTG devices must have a mini-A plug on one end and a mini-B plug on the other end, and it doesn't matter which device has which plug. A cable that connects an OTG device and a peripheral-only device has a mini-A plug on one end, and the other end may have a B plug, a mini-B plug, or for captive cables, a vendor-specific connector or permanent attachment to the device.

The On-The-Go supplement allows cable adapters on devices with captive cables. To attach to a host or hub with A receptacles, a device with a mini-A plug on a captive cable can use an adapter that has a mini-A receptacle and a standard A plug. To attach to an OTG device, a device with an A plug on a captive cable can use an adapter that has a standard-A receptacle and a mini-A plug. These are the only approved cable adapters and the only approved use for the mini-A receptacle. To allow the use of adapters, all cables with mini-A connectors must have slightly shorter propagation delays (25 nanoseconds maximum) and a maximum length of 4.5 meters.

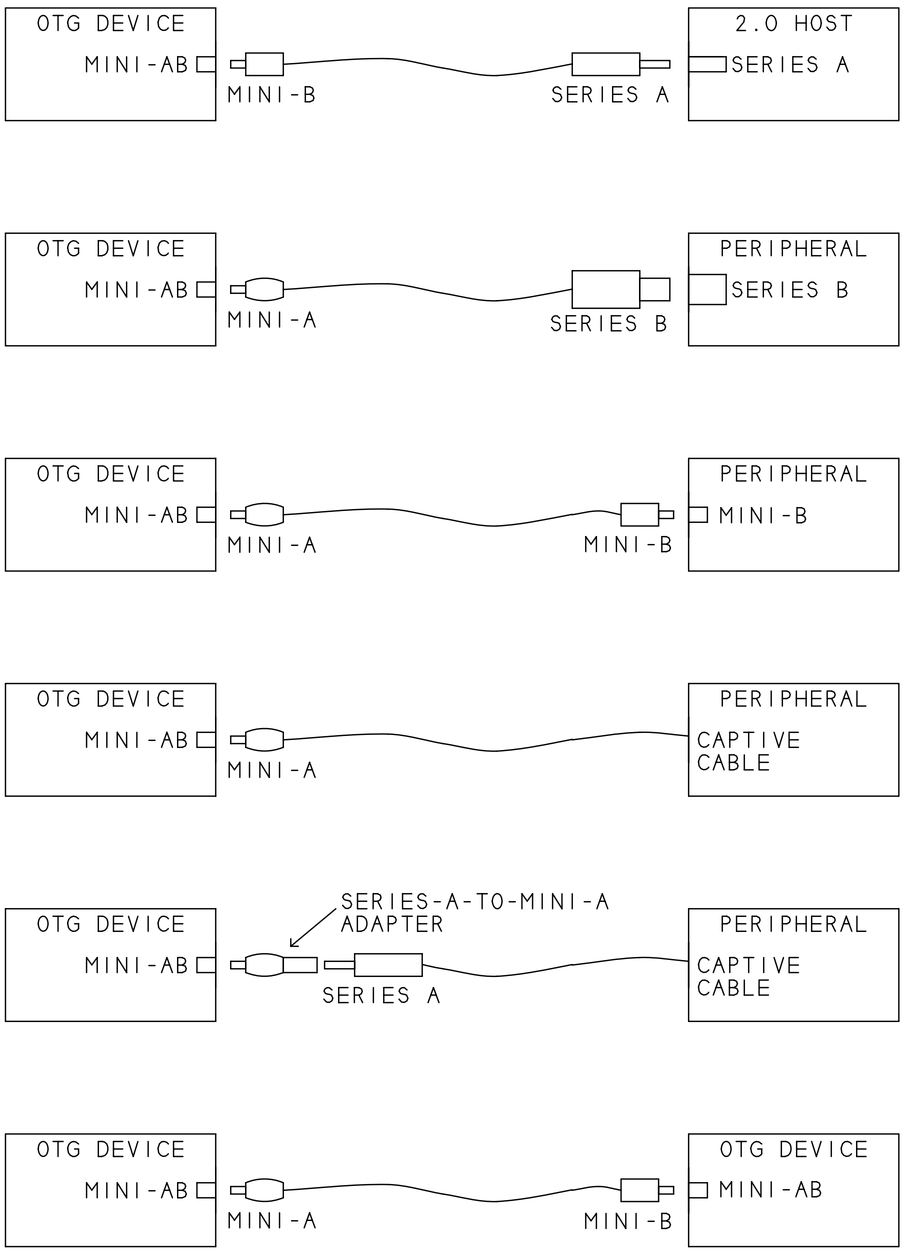


Figure 20-2: An OTG device might use any of these cable types to connect to hosts, peripherals, and other OTG devices.

In addition to D+, D-, VBUS and GND, the mini-A, mini-B, and mini-AB connectors have an ID pin. This pin enables an OTG device to determine whether a mini-A or mini-B plug is attached. In a mini-A plug, the ID pin is grounded. In a mini-B plug, the ID pin is open or connected to ground via a resistance greater than 100 kilohms. An OTG device typically has a pull-up resistor on the ID pin. If the pin is a logic low, the attached plug is a mini-A, and if the pin is a logic high, the attached plug is a mini-B.

### Bus Current

The ability to draw up to 500 milliamperes from the bus is a convenience for users and a cost saver for device manufacturers. But providing this much current, or even the 100 milliamperes that battery-powered hosts must provide, can be a burden for an OTG device. And some peripherals, including battery-powered ones, may not need bus power at all.

For these reasons, an OTG device is required to supply only 8 milliamperes of bus current. OTG devices that need to supply more current to their peripherals are free to do so, up to 500 milliamperes.

Many OTG devices will need to supply more than 8 milliamperes. For example, a keyboard with a few LEDs could easily require 50 milliamperes from the host. A device whose targeted peripheral list includes an entire class (or a HID Usage, such as keyboards) should be sure that sufficient current is available to power any such device that users might attach.

### User Messages

To prevent user frustration, every On-The-Go device must include and use a display or another way to communicate messages to users. For example, if a user connects an unsupported printer to a dual-role camera, a message of "Device not supported" would be helpful.

### The Targeted Peripheral List

Every OTG device must have a targeted peripheral list that names all of the devices the host can communicate with. The On-The-Go supplement doesn't specify where the list must appear. Users will appreciate it if the information is easy to find!

## The OTG Descriptor

During enumeration, a device that supports HNP or SRP must include an OTG Descriptor in the descriptors returned in response to a Get_Descriptor request for the Configuration descriptor.

Table 20-2 shows the fields of the descriptor. The bmAttributes field tells whether the device supports HNP and SRP. A device that supports HNP must support SRP. The A-device doesn't need to know in advance if a device supports SRP, but this information is included in the descriptor for use in compliance testing.

## Feature Codes for HNP

The OTG supplement defines three codes for use in Set_Feature requests.

A code of b_hnp_enable (03h) informs the B-Device that it can use HNP. The A-device sends this request if all of the following are true: the A-device supports HNP, the A-device will respond to HNP when the bus is suspended, and the B-device connects directly to the A-device, with no hubs in between.

A code of a_hnp_support (04h) informs the B-device that the A-device supports HNP and the B-device is directly connected (no hubs), but the B-device isn't yet allowed to use HNP. The A-device can send this request before configuring the B-device. The A-device can then enable HNP at a later time, when the A-device is finished using the bus.

A code of a_alt_hnp_support (05h) notifies the B-device that the currently connected port does not support HNP, but that the A-device has an alternate port that does support HNP.

# OTG Controller Chips

Several manufacturers offer controller chips designed for use in OTG devices. To function as a peripheral, the controller must include device-controller circuits similar to those in the controllers described in Chapter 6. As

Table 20-2: The OTG Descriptor indicates whether a device supports HNP and SRP.

| Offset | Field | Size | Description |
|---|---|---|---|
| 0 | bLength | 1 | Descriptor length (3) |
| 1 | bDescriptorType | 1 | OTG (9) |
| 2 | bmAttributes | 1 | D2–D7: reserved,<br>D1: 1 = HNP supported,<br>0 = HNP not supported<br>D0: 1 = SNP supported,<br>0 = SNP not supported |

with other device controllers, some OTG device controllers contain a CPU while others must interface to an external CPU.

To function as an OTG device, the controller (possibly with the help of external circuits) must have the ability to send SOF packets, schedule and initiate Setup, IN, and OUT transactions, provide VBUS, manage power, reset the bus, switch the pull-up and pull-down resistors as needed when changing roles, and detect the state of the ID pin. Some chips have internal charge pumps for supplying and controlling VBUS from a 3V supply.

A controller may also provide timers, status signals, or other hardware support for SRP and HNP signaling.

## Philips ISP1362

Philips Semiconductor's ISP1362 is an interface-only chip for OTG devices. The chip contains an ISP1181B device controller (described in Chapter 6) and a host controller. Both controllers can communicate at full and low speeds. (The OTG device must use full speed when functioning as a peripheral.)

The controller interfaces to an external CPU using a 16-bit interface that can transfer data at up to 10 Megabytes/sec. The external CPU communicates with the controller by accessing its registers and buffer memory. The registers are compatible with the registers defined in the OHCI specification.

The Philips Transfer Descriptor (PTD) defines a format for exchanging information with the host controller's driver. The descriptor consists of a header that contains information such as the endpoint number, transaction type (Setup, IN, OUT), bus speed, toggle-bit value, and a completion code, followed by data.

The chip contains two USB ports. One port can function as the OTG port in a OTG device or as a host or device port for a non-On-The-Go host or device. The second port can function only as a host port and is not recommended for use in On-The-Go devices.

Philips provides host, peripheral, and OTG drivers for PCI platforms running Linux, Windows CE, DOS, and the FlexiUSB real-time operating system and for Intel PXA250/Arm architecture platforms running Linux or Windows CE.

If you need high speed, the ISP1761 is an OTG controller that supports high speed and can use a 16-bit or 32-bit CPU interface.

## TransDimension TD242LP

Transdimension Inc.'s TD242LP is a physically small, low-power interface chip especially suited for compact and inexpensive dual-role products.

The controller interfaces to an external CPU using a 16-bit data bus that can transfer data at up to 22 Megabytes/sec.

The chip has two USB ports that can be configured in any of four modes. In Hardware HNP mode, one port is an OTG device port, the other is a non-OTG host port, and HNP is handled in hardware. Software HNP mode is the same except that HNP is handled in software. In Host Only mode, there is a single non-OTG host with two ports. In Function Host mode, one port is for a non-OTG host and the other is for a peripheral-only device. Both ports can operate at low and full speeds.

For reduced EMI, the chip can be clocked at 6 Megahertz rather than 48 Megahertz.

The Host Endpoint Transfer Descriptor defines a format for sending and receiving USB data. Information in the descriptor includes the endpoint number, transfer type, bus speed, direction, and a completion code.

Transdimension supplies a host-controller driver for Linux and Windows CE. Other operating systems can use USBLink drivers from SoftConnex.

## Cypress CY7C67200 EZ-OTG

As the name suggests, Cypress Semiconductor's CY7C67200 EZ-OTG controller is designed for use in OTG devices. The chip contains a 16-bit CPU and can function in two modes. In stand-alone mode, the controller is the device's main CPU. Firmware can be stored in an $I^2C$ EEPROM or the controller can download its firmware from a USB host using the same method used by the EZ-USB chips described in Chapter 6. In coprocessor mode, the controller interfaces to an external CPU that manages USB communications and other tasks. The CPU can communicate via either a parallel Host Peripheral Interface at up to 16 Megabytes/sec., a high-speed asynchronous serial interface at up to 2 Megabaud, or a Serial Peripheral Interface (SPI) at up to 2 Megabits/sec.

The EZ-OTG has two USB ports and two serial interface engines that support low and full speeds. One port can function as an OTG device, a non-OTG host, or a peripheral-only device port. The other port can function as a non-OTG host or peripheral-only device port.

The controller contains a ROM BIOS that executes an Idle task consisting of an endless loop that waits for an interrupt, executes the tasks in the Idle chain, and repeats. Firmware can add tasks to the Idle chain or replace the entire Idle task with device-specific programming.

Firmware development can use the free GNU Toolset, which supports many CPUs and includes a C compiler, assembler, make utility, linker, debugger and other utilities. Cypress provides Frameworks C code for performing USB-related tasks and accessing other components in the controller.

A tutorial and many examples are in the free e-book, *USB Multi-Role Device Design By Example*, by John Hyde, available from *www.usb-by-example.com*.

A related chip, the CY7C67300 EZ-HOST, includes an interface to external memory, two ports for each of the two SIEs, memory expansion capabilities, and additional I/O features.

## Philips ISP1261 Bridge Controller

Philips Semiconductor's ISP1261 is a bridge controller that takes a different approach to OTG design. With this controller and some additional firmware, an ordinary USB device can function as a "pseudo host" that can communicate with USB devices.

The ISP1261 adds some overhead, so it doesn't provide the most efficient communications, but the chip can offer a quick way to add host capability to a device. The controller can be integrated into a device or implemented as a separate dongle that attaches to a device's USB port. When implemented as a dongle, the bridge requires no hardware changes on the device, though the device must be able to store and run new firmware that communicates with the bridge controller.

The bridge contains a host port and host controller and an OTG port and OTG controller. The host port connects to the local device, and the OTG port connects to the remote host or device that the local device wants to communicate with. The host controller communicates with the local device, and the OTG controller communicates with the remote device. A Software Emulated OTG Controller (SEOC) Protocol Engine manages communications between the host controller and the OTG controller. A state machine implements the SEOC Protocol.

***

I hope you've found *USB Complete* to be useful. For more about USB developing, including device and host example code and links to product information, tutorials, articles, news, and updates, please visit my Web site at *www.Lvr.com*.

Jan Axelson