Simplifying Universal Serial Bus Connectivity with USB Bridge Devices

Ramya Kota, Product Manager, Silicon Labs

The universal serial bus (USB) interface is one of the most successful communication standards in the electronics industry. It has become the standard communication interface for many industrial and consumer applications that require connectivity to a host device. USB’s extreme ease of use coupled with robustness makes it the ideal choice of interface for many embedded applications.

However, for embedded designers, a USB implementation requires USB protocol expertise, time-consuming software and firmware development effort. Most often, designers of these embedded applications are required to provide an end solution that is highly cost effective. The simplest way to achieve this is by integrating the USB functionality in the microcontroller (MCU) along with other peripherals and memory.

Many MCU vendors today offer USB-based microcontrollers; however, they may not offer the right peripheral set required for an application, resulting in the need for external components and increased design complexity. There’s also a no-hassle approach for adding USB to any embedded solution, with no need for USB expertise or firmware development, enabling faster times to market for designers.
USB Design Challenges

The ease of use and robustness of USB do not come for free for embedded designers. Developers often have to spend significant time learning the USB protocol and developing a USB stack. This requires specialized tools and can increase development costs and time. There are commercially available USB software stacks; however, they result in additional costs and learning effort. Another important USB design challenge is the fact that the final product must be compatible with several common operating systems (OS) and keep up with constant OS updates. Developers also must keep the USB hardware costs to a minimum to provide a cost-effective final product. Many USB ICs require external components such as termination resistors or precision oscillators to provide USB functionality. Highly integrated USB solutions with advanced software and tools support are the cornerstone to simplifying USB design.

The Practical Fix: Turnkey USB Connectivity Solutions

Turnkey USB solutions, such as Silicon Labs’ USBXpress bridge devices, provide high integration to eliminate the need for complex firmware and driver development. USB devices with advanced tools support also reduce development time, further simplifying USB connectivity. Highly integrated USB bridge devices can be added to virtually any MCU-based solution to eliminate firmware complexity and reduce development time. These devices act as a bridge to the USB world through the use of standard UART, SPI or I2C interfaces that are readily available on most MCUs.

Bridge devices are especially helpful when developers are upgrading a legacy system with USB communications. Most legacy designs have a specific means of communicating with the host system, in many cases through a UART or I2C interface. A USB-to-UART or USB-to-I2C bridge device, shown below, is the best option in this case.

---

The ease of use and robustness of USB do not come for free for embedded designers. Developers often have to spend significant time learning the USB protocol and developing a USB stack.

---

Choosing the Right USB Bridge for Your Application

Embedded designers should consider a multitude of factors when choosing a USB bridge device to ensure seamless USB integration.

First, it is important to minimize hardware design cost that can easily impact an allotted budget. Most USB bridges or MCUs require external components, such as crystal oscillators or termination resistors for USB functionality.
Choosing a highly integrated bridge device can eliminate the need for external components, thereby reducing hardware design time and cost.

For a full-speed function, the USB device requires a 48 MHz clock input. Choosing a USB device with an internal clock that can generate a 48 MHz clock eliminates the need for an external crystal. Another important feature to look for is the clock recovery circuitry. The USB specification requires the internal clock to be highly accurate. However, most often, the frequency accuracy of the internal oscillators can vary. The clock recovery circuitry uses the incoming USB data stream to adjust the internal oscillator, and it allows the internal oscillator to meet the requirements for USB clock tolerance. Devices that use this novel clock recovery system reduce costs by eliminating the need for an external crystal oscillator.

One of the major advantages of adding USB to embedded applications is the ability to power devices. Battery-powered devices can maximize the battery life when the USB device is bus-powered while connected to the host. This requires a low drop-out (LDO) or voltage regulator to drop the voltage from 5 V (host) to 3.3 V. It is also essential to use a USB function controller with an integrated transceiver and on-chip matching and pull-up resistors. The integration of the termination resistors for proper USB line termination and a 5 V regulator for direct USB connection further reduces the size of the printed circuit board (PCB).

Additionally, USB bridges offering GPIO control capability and clock output functions add flexibility to designs and shorten development times. Another attractive feature to consider in bridge devices is the flexibility to support multiple baud rates to fit a wide range of applications.

Some modern USB bridge devices also offer advanced features to further conserve the PCB area. For example, the CP2102N device from Silicon Labs includes an integrated battery charger detection feature that detects the type of charger connected and a remote wake-up feature to wake up a suspended host. Inclusion of such advanced features on-chip not only enhances flexibility for embedded designs but also saves overall system cost.

Finally, designers must minimize USB space to meet the form factor and ergonomic requirements of space-sensitive portable applications. In addition, some legacy applications may require keeping the same physical dimensions while upgrading the application to include USB connectivity. Choosing USB bridges that offer small package variants helps overcome this constraint by reducing the PCB footprint and enabling the designer to squeeze a solution into a space smaller than an RS-232 connector.
Accelerated USB Development with Advanced Tools

High integration is a critical factor that determines the selection of a USB bridge to reduce costs and hardware design time, but also equally important is the software and tool set in making the right bridge choice. Consumers view USB as the perfect plug-and-play interface. While it is true that USB looks as simple as other serial interfaces such as UART, SPI or I2C, the USB protocol is not simple to implement. This hidden complexity of USB protocol and firmware development for designers can be solved by using turnkey fixed-function USB bridge devices. These devices allow designers to develop USB systems without the need for extensive USB expertise or firmware development.

While there is no code development required when using fixed-function devices, designers are still looking for several configuration options to suit their designs or to differentiate their designs in the market. Each USB bridge device has many user-configurable options that determine the device’s behavior, ranging from how the device will appear to the host machine to specialized device options such as interface and pin configurations. Therefore, it is essential to make sure that the selected USB bridge devices offer advanced tools that provide configuration options and allow easy customization to differentiate the end products.

The Xpress configurator from Silicon Labs is an example of a tool designed to simplify USB configuration. The intuitive GUI-based configurator allows easy customization and programming of USB fixed-function devices. Common USB attributes such as Vendor ID (VID), Product ID (PID), device strings and USB power modes can be configured easily using the GUI. Advanced features such as GPIOs, interface options and modem signals can also be configured with just a few clicks. Additionally, having a problems/warnings view in the GUI guides developers toward the settings for each USB attribute and provides a more risk-free environment for USB customization. The final desired configuration can be programmed using a single button to use turnkey devices without writing any code.

Most often, developers do not have the right tool set for programming thousands of PCBs and devices with the final configurations. In such cases, it is extremely useful to select USB device vendors that offer factory programming to save significant development times for the developers.

Example of an advanced tool GUI for USB bridge configuration
Software Drivers

A USB device is expected to work seamlessly across all operating systems. To achieve this goal, developers must create host drivers, which is not a trivial task. USB driver development requires expertise and significant testing for the best operation. It is essential to choose USB devices that offer robust certified drivers for seamless USB connectivity in systems. Some vendors offer Virtual COM Port (VCP) device drivers that allow the devices to appear as a COM port to the PC’s application software. This is especially useful when trying to update legacy serial devices with USB. With the VCP drivers, the PC software recognizes the device as a COM port, and hence existing COM port applications may be used to transfer data via USB without actually modifying the application.

To further simplify and enhance plug-and-play capabilities for designers, a few USB device vendors offer innovative, fully-compliant USB-HID bridges. These devices support the USB-HID class that is natively supported by most operating systems, thereby eliminating the need for driver installation or development. The most common examples of HID-USB devices are computer keyboards and mice, and the HID-USB class is also flexible enough to accommodate many different kinds of USB designs.
Conclusion

USB bridges and turnkey USB solutions provide an easy and economical approach to add USB connectivity to new or existing non-USB systems. Their high integration reduces PCB size and BOM costs, making them ideal for space-constrained and cost-sensitive applications. Their simplicity and turnkey nature eliminate the need for USB expertise or firmware development, enabling designers to get their end products to the market faster. Advanced USB configuration tools and robust certified driver packages allow developers to differentiate their applications and provide seamless compatibility with most operating systems. It is essential to make USB easy for not only consumers but also for designers, and choosing the right USB bridge device will take the hassle out of adding USB connectivity to your next embedded design.