Typische Entwicklungsfehler Erkennen Und Vermeiden

Von (Steve Kolokowsky, Semiconductor;s and Computatuin Divisioin, Cypress Semiconductor Corp.)

Zusammenfassung

Seit der ersten Einführung des Universal Serial Bus (USB) sind mittlerweile elf Jahre vergangen. Ob man es glaubt oder nicht – Tag für Tag werden neue und innovative Möglichkeiten zur Nutzung dieses Protokolls gefunden. Ebenso werden täglich überall auf der Welt dieselben Fehler gemacht. Das müsste nicht sein.

Interessanterweise weist der USB viele Ähnlichkeiten mit anderen Protokollen auf, die dem Entwickler möglicherweise bereits vertraut sind. Trotzdem gibt es so viele Unterschiede zum bekannten Terrain von PS/2 und RS-232, dass die Ingenieure immer wieder über dieselben Fallstricke stolpern. Wegen der strikten Standards können diese Fehler im schlimmsten Fall dazu führen, dass die USB-Compliance nicht eigehalten wird. Man kann die Fehler beim USB-Design in fünf Kategorien einteilen: Übertragungsgeschwindigkeit, Stromversorgung, Signalqualität, Software und Compliance.

Seit der ersten Einführung des Universal Serial Bus (USB) sind mittlerweile elf Jahre vergangen. Ob man es glaubt oder nicht – Tag für Tag werden neue und innovative Möglichkeiten zur Nutzung dieses Protokolls gefunden. Ebenso werden täglich überall auf der Welt dieselben Fehler gemacht. Das müsste nicht sein.

Interessanterweise weist der USB viele Ähnlichkeiten mit anderen Protokollen auf, die dem Entwickler möglicherweise bereits vertraut sind. Trotzdem gibt es so viele Unterschiede zum bekannten Terrain von PS/2 und RS-232, dass die Ingenieure immer wieder über dieselben Fallstricke stolpern. Wegen der strikten Standards können diese Fehler im schlimmsten Fall dazu führen, dass die USB-Compliance nicht eigehalten wird. Man kann die Fehler beim USB-Design in fünf Kategorien einteilen: Übertragungsgeschwindigkeit, Stromversorgung, Signalqualität, Software und Compliance.

1. Übertragungsgeschwindigkeit:

   - Mögliche Engpässe im System werden nicht einkalkuliert

2. Stromversorgung

   - Kenntnis der Bus-Stromversorgung (Grenzwerte von 500 mA, 100 mA, 500 µA)
   - Stromrückfluss auf D+
   - Hubs: Keine volle Versorgungsspannung an Downstream-Ports
   - Busgespeiste Hubs: Umschalten zwischen hoher und niedriger Leistungsaufnahme

3. Signalqualität

   - Nutzung von D+/D- durch mehrere Devices
   - Schalter im Datenpfad

4. Software

   - Keine Verwendung verfügbarer Klassentreiber
   - Kein Einholen einer Vendor ID (VID) vom USB I/F

5. Compliance

   - Kein Testen des Prototyps vor dem Einreichen zur Compliance-Prüfung
   - FCC/VCCI (EMI): Keine Trennung von Abschirmung/Masse

Published in Elektronik Praxis (http//: www.elektronikpraxis.de) March 2007

Fehler bei der Übertragungsgeschwindigkeit vermeiden

Die Frage, welche Übertragungsrate ein USB-Device unterstützt, ist zweifellos die Ursache für die meisten Missverständnisse zwischen Entwicklern und Anwendern. Sie ist überdies die meistgestellte Frage in Computerläden auf der ganzen Welt. Der leitungsgebundene USB unterstützt derzeit drei Übertragungsraten, nämlich Low Speed (1,5 MBit/s), Full Speed (12 MBit/s) und High Speed (480 MBit/s). Allerdings ist Vorsicht geboten: USB 2.0 ist nicht gleichbedeutend mit High Speed USB, auch wenn High Speed USB erstmals mit der Einführung von Version 2.0 der USB-Spezifikation eingeführt wurde. Auch USB 2.0 unterstützt Low Speed und Full Speed.

Wie bei allen elektronischen Systemen, so gilt auch hier, dass der Designer die größtmögliche Performance anstrebt. Leider gehen in USB-Applikationen die meisten Designer fälschlicherweise davon aus, dass sie mit ihrem System die volle Übertragungsrate von 1,5, 12 oder 480 MBit/s erreichen werden. Mehrere Gründe sind ausschlaggebend dafür, dass ein reales System niemals auf diese Bandbreite kommt.

Mehrere Geräte teilen sich einen Host-Controller

Erstens wird der USB von allen angeschlossenen Geräten gemeinsam genutzt. Selbst wenn man jedes Gerät mit einem eigenen USB-Anschluss des Computers verbindet, ist es sehr wahrscheinlich, dass man dabei ein und denselben USB Host Controller nutzt. Jedes Gerät muss sich somit die Gesamt-Bandbreite mit allen übrigen Devices teilen.

Zweitens handelt es sich bei USB um ein paketorientiertes Protokoll. Längere Datenblöcke werden also in Pakete zu jeweils 512 Bytes zerlegt. Jedes Paket wird mit einem Header versehen, der Auskunft über den Inhalt gibt, und erhält am Schluss einen CRC-Code, mit dem die Integrität der enthaltenen Daten bestätigt wird. Um das korrekte Timing des Busses zu gewährleisten, wird außerdem alle 125 µs ein SOF-Paket (Start of Frame) gesendet (Microframe). Dies führt unter dem Strich dazu, dass die theoretische Maximalbandbreite des USB 13 Bulk-Pakete pro Microframe, das sind 53.248.000 Bytes/s, ausmacht. Doch auch diese Marke ist mit den gegenwärtigen Host-Controllern, die 10 Bulk-Pakete je Microframe empfangen bzw. 8 Bulk-Pakete je Microframe senden können, nicht erreichbar.

Mögliche Engpässe im System werden nicht einkalkuliert

Diesem Fehler ist bereits mehr als ein System zum Opfer gefallen. So analysierte man bei Cypress Semiconductor unlängst ein System, das Daten in einen NAND-Flash-Speicher schrieb. Das System sendete Daten über den USB, legte diese Daten in einem Pufferspeicher ab und schrieb sie von dort in den Flash-Speicher. Für jedes Paket kamen drei Zeitkomponenten zum Tragen: 1.) Die Zeit zum Ausführen des USB-Transfers, 2.) der primäre, auf das Betriebssystem zurückzuführende Zeitaufwand und 3.) die Programmierzeit der NAND-Flash-Firmware. Eine Echtzeit-Analyse der Performance ergab im Falle eines 128-KByte-Blocks das folgende Timing:

Bis zu dieser Analyse hatten die Ingenieure alles daran gesetzt, durch Beschleunigung der Signale für den USB-Interfacechip die Übertragungszeit auf dem USB zu verkürzen. Sobald klar war, dass die Programmier-Firmware des NAND-Flash den weitaus größten Einfluss auf die Leistungsfähigkeit hatte, konnten sie sich stattdessen darauf konzentrieren, die Performance durch Reduzieren des Flash-Overheads zu steigern. In den meisten Systemen wird die High-Speed USB-Übertragung nicht der Engpass sein. Die Designer müssen deshalb ihren Blick auf das Gesamtsystem richten und sicherstellen, dass die für die angestrebte Systemgeschwindigkeit erforderliche Bandbreitenreserve vorhanden ist.

Fehler bei der Stromversorgung über den USB

Gemäß der USB-Spezifikation kann ein USB-Device entweder über das Buskabel mit Strom versorgt werden (‚Bus-Powered') oder über eine eigene dezentrale Stromversorgung durch Batterie oder Netzteil verfügen (‚Self-Powered'). Die Möglichkeit der Stromversorgung über das Buskabel war eine der besten Ideen bei der Entwicklung des USB, kann doch auf ein separates Netzkabel verzichte Zu Problemen kann es jedoch auch bei USB-Devices mit eigener Stromversorgung kommen, da diese Geräte eingeschaltet sein können, während der Host abgeschaltet ist. Problematisch ist es, wenn die geringe Pull-up-Spannung, die an D+ gelegt wird, um die Erkennung von USB-Devices zu ermöglichen, langsam das gesamte Hostsystem auflädt und den Startup-Vorgang stört. USB-Geräte mit eigener Stromversorgung (auch solche mit Batterie) müssen diese Pull-up-Spannung deshalb direkt aus VBUS beziehen oder sie softwaregesteuert mittels eines VBUS-Sensors abschalten. t werden. Allerdings müssenbei der Stromversorgung per USB die von der USB-Spezifikation gesetzten Grenzen von 500 uA, 100 mA und 500 mA beachtet werden. Viele Designer achten nicht sorgfältig genug auf die Einhaltung dieser Grenzwerte und

bauen Systeme, die den Regeln für Bus-Powered Devices nicht genau entsprechen. Wie kommt es zu den drei verschiedenen Strom-Limits?

- 500 µA – Wenn der Host eine Betriebsspannung zur Verfügung stellt, aber der USB inaktiv ist, muss sich das Device im Suspend-Modus befinden, in dem es der VBUS-Leitung höchstens 500 µA entnehmen darf. Diese Betriebsart soll die Stromaufnahme von PCs im Suspend-Modus minimieren.
- 100 mA – Am USB gibt es High-Power- (500 mA) und Low-Power-Ports (100 mA). Low-Power-Ports findet man in der Regel an busgespeisten Hubs, die dem USB 500 mA entnehmen und jeden ihrer Downstream-Ports mit maximal 100 mA versorgen können. Beim ersten Einstecken eines Device in einen Port weiß das Gerät nicht, welcher Art der Port ist. Es ist somit auf 100 mA beschränkt, bis es eine SET_CONFIGURATION-Meldung vom Host erhält. Ein Gerät muss also so konzipiert sein, dass es den Enumeration-Prozess am USB mit sehr geringer Leistungsaufnahme abwickeln kann, bis es nach dem Erhalt der SET_CONFIGURATION-Meldung in eine Betriebsart mit höherer Leistungsaufnahme wechselt. Dies gestaltete sich im Zusammenhang mit High-Speed USB als schwierig, bis Cypress Semiconductor im Jahr 2004 seinen FX2LP-Chip einführte.
- 500 mA – Dies ist die nach den USB-Spezifikationen maximal zulässige Leistungsaufnahme.

Systementwickler sollten unbedingt Tests unter Praxisbedingungen durchführen, um sicherzugehen, dass sie die verschiedenen Verlustleistungs-Stufen für den busgespeisten Betrieb einhalten. Sonst könnte es passieren, dass das schöne neue USB-Design mit einem teuren Steckernetzteil ausgeliefert werden muss.

Stromrückfluss in den USB

Zu Problemen kann es jedoch auch bei USB-Devices mit eigener Stromversorgung kommen, da diese Geräte eingeschaltet sein können, während der Host abgeschaltet ist. Problematisch ist es, wenn die geringe Pull-up-Spannung, die an D+ gelegt wird, um die Erkennung von USB-Devices zu ermöglichen, langsam das gesamte Hostsystem auflädt und den Startup-Vorgang stört. USB-Geräte mit eigener Stromversorgung (auch solche mit Batterie) müssen diese Pull-up-Spannung deshalb direkt aus VBUS beziehen oder sie softwaregesteuert mittels eines VBUS-Sensors abschalten.

Fehlerhafte Signalqualität durch Mehrfachnutzung der Signalleitung

Um Zeit, Aufwand und Geld zu sparen, werden bei einigen Produkten mehrere Devices an die USB-Signalleitungen angeschlossen. Zum Beispiel kann bei einer USB-Docking-Station die Möglichkeit vorgesehen sein, wahlweise ein Floppy- oder DVD-Laufwerk in einen Massenspeicher-Slot einzusetzen. Diese Art der Mehrfachnutzung ist zweifellos kostensparend, da weniger Hub-Ports benötigt werden. Es ist aber schwierig, eine dieser Lösungen zu implementieren, ohne die Charakteristika sämtlicher Devices im System vollständig verstanden zu haben. In der Tri-State-Konfiguration (Bild 4, Option 2) führt das andere am Bus angeschlossene Device zu einer Erhöhung der Kapazität an den USB-Leitungen. Hinzu kommt, dass die zu diesem weiteren Device führende Leiterbahn Signalreflexionen verursacht, die den High-Speed USB-Betrieb stören kann. In der Anordnung nach Bild 4, Option 1 erhöht der Schalter sowohl die Kapazität als auch den Widerstand der USB-Leitungen, was die Anstiegs- und Abfallzeiten der USB-Leitungen erhöht und das Auge der USB-Signale enger macht.

Das Augendiagramm des von einem USB-Device gesendeten Signals nach Durchlaufen eines Schalters mit 10 pF/10 □. Es dürfte zu keinen Interferenzen mit dem roten Bereich kommen. Entscheidend für die erfolgreiche Mehrfachnutzung von USB-Signalleitungen ist es, die Zusatzbelastung gering zu halten und Chips mit hoher Flankensteilheit zu verwenden.

Durch Software verursachte Fehler

Innerhalb des USB-Ökosystems sind die Klassentreiber ein wichtiges Element. Sie sind in den wichtigen Betriebssystemen bereits enthalten, sodass die Entwicklung eines eigenen Gerätetreibers nicht erforderlich ist. Die USB-Klassen werden von Device Working Groups definiert, die auf freiwilliger Basis an der Schaffung einer Standardsprache für den Dialog mit USB-Devices arbeiten. Bei den heute bestehenden USB-Geräteklassen handelt es sich um HID (Human Interface Device) mit Mäusen, Tastaturen und anderen Eingabegeräten, Mass Storage (Disklaufwerke), Communication (Modems, Netzwerkadapter), Audio, Video, und Still Image (Fotokameras und Scanner).

Standard-Klassentreiber sind meist hilfreich

Wenn Ihr Device genau in die existierende Klassenstruktur fällt, ist die Entscheidung einfach, denn in diesem Fall muss nur unter www.usb.org die entsprechende Klassendefinition heruntergeladen und implementiert werden. Doch auch wenn das Device nicht genau in eine existierende Klasse passt, können diese Standard-Klassentreiber hilfreich sein. Zum Beispiel verwendete Microsoft die Klasse Still Image zur Implementierung der neuen Klasse MTP (Media Transfer Protocol). Auch die HID-Klasse muss nicht unbedingt mit einer Funktion verbunden werden, die mit einem Benutzer interagiert, sondern lässt sich auch verwenden, um beispielsweise Thermometer, Drucksensoren oder Pumpensteuerungen ohne zusätzliche Treiber mit einem Softwareprogramm zu verbinden.

Aus irgendwelchen Gründen versuchen viele Unternehmen, eigene Treiber mit großem Kostenaufwand entweder selbst zu entwickeln oder die Entwicklung an ein externes Designhaus zu vergeben, obwohl die Implementierung eines vorhandenen Klassentreibers ausreichend wäre. Immerhin eliminieren Klassentreiber Designrisiken, senken die Kosten und vermeiden mögliche Zeitplanüberschreitungen. Auch Debugging- und Komplexitätsprobleme gehören der Vergangenheit an.

Keine Vendor-ID (VID) vom USB IF eingeholt

Jedes USB-Device ist mit einer eindeutigen Kennung versehen, die es dem Betriebssystem ermöglicht, den richtigen Treiber auszuwählen. Den ersten Teil dieses ‚Identifiers' bildet ein 16-Bit-Wert, der vom USB Implementor's Forum (www.usb.org) vergeben wird (VID). Der zweite Teil ist ein 16-Bit-Wert, dessen Vergabe dem jeweiligen Anbieter (d. h. dem Entwickler des Produkts) überlassen bleibt (PID).

Das Einholen der VID und die Vergabe der PID sind eine einfache Sache, wenn das Unternehmen und seine Designer vorausschauend planen und gut mit dem USB Implementers Forum kommunizieren. Leider kommen jedoch alljährlich Tausende von Produkten verspätet auf den Markt, weil während der Arbeit an der Software und der Firmware noch keine Kennungen vorliegen und VID und PID für eine einwandfreie Identifikation benötigt werden.

Fehler durch mangelhafte Compliance

Voraussetzung für die legale Verwendung der verschiedenen USB-Logos ist das Bestehen einer USB-Compliance-Prüfung. Das USB IF führt solche Tests durch, um zu gewährleisten, dass die Endkunden beim Einsatz von USB-Produkten nicht mit Problemen konfrontiert werden. Schließlich verlassen sich sämtliche USB-Anbieter darauf, dass im Sinne der Benutzer zusammengearbeitet wird. Sollte ein Kunde Probleme mit einem USB-Gerät haben, wird er nach dieser negativen Erfahrung weniger bereit sein, Zeit und Geld in ein weiteres USB-Gerät zu investieren.

Einige der Tests, die im Zuge der Compliance-Prüfungen durchgeführt werden müssen, erfordern anspruchsvolles Test-Equipment, das sich der einzelne Entwickler unter Umständen nicht leisten kann. Viele Devices fallen jedoch wegen ganz simpler Dinge, die jeder Entwickler testen kann, durch die Prüfung. Vor einem Plugfest oder bevor Sie ein Gerät zum Compliance-Test einreichen, sollten Sie unbedingt die folgenden Prüfungen vornehmen:

- Chapter 9 Test – USB Command Verifier Tool. Dieses Programm überprüft, ob Ihr Gerät die wichtigsten von einem Host ausgegebenen SETUP-Befehle ausführen kann (Tools siehe unter http://www.usb.org/developers/tools/).
- Power Tests – geprüft werden der Suspend- und der Inrush-Strom und die Stromaufnahme im unkonfigurierten Zustand.
- EMI: Über gute EMI-Designtechniken könnte man einen eigenen Artikel schreiben. Der häufigste und gleichzeitig am einfachsten zu behebende Fehler im Zusammenhang mit EMI-Abwehrmaßnahmen ist die direkte Verbindung der Abschirmung des USB-Kabels mit der Ground Plane des eigenen Systems. Jegliche in die Ground Plane eingestreuten Störungen können dadurch der Abschirmung um Ihr Gerät herum entkommen, ganz zu schweigen davon, dass die Störungen jetzt eine 2 m lange Antenne vorfinden!

Die eben geschilderten Problembereiche bereiten Hunderten von Ingenieuren jedes Jahr Kopfzerbrechen. Das müsste nicht sein. Nacharbeiten und daraus resultierende Verzögerungen lassen sich vermeiden, wenn man aus den Fehlern Anderer lernt. Stellen Sie jedoch keinesfalls Ihre Kreativität ein, denn die Welt braucht auch in Zukunft immer innovativere, ausgefeiltere und komfortablere USB-Geräte.

.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
http://www.cypress.com