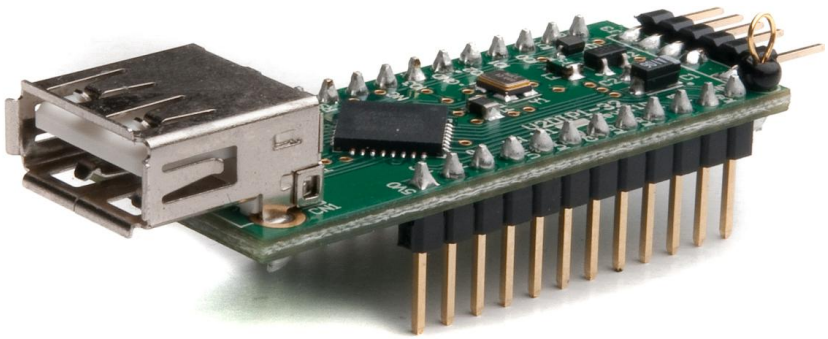
DESIGN GUIDE

USB Wechseldatenträger-unterstützung

IMPLEMENTIERUNG FÜR EINGEBETTETE SYSTEME



Datakey®

The power of memory. Secured.

Datakey
The power of memory. Secured.

Whitepaper

Design Guide für eingebettete Systeme mit USB-Wechseldatenträger-Unterstützung

ZUSAMMENFASSUNG

Ein eingebettetes System mit USB-Wechseldatenträger-Unterstützung zu implementieren, kann sich als schwieriger erweisen als gedacht. Wer die wichtigsten Konzepte und Überlegungen kennt, hat es hier viel leichter.

Inhaltsverzeichnis

Einleitung

USB-Wechseldatenträger zählen in der Welt des Embedded Computing zu den Schlüsseltechnologien. Sie können nicht nur genutzt werden, um Daten darauf zu speichern, sondern beispielsweise auch, um Firmware automatisch zu aktualisieren oder um sicheren, authentifizierten Zugriff zu gestatten. USB-Wechseldatenträger heben das Embedded Computing auf ein neues Niveau.

Während die Verwendung von USB-Wechseldatenträgern ganz einfach ist, kann sich ihre Implementierung als schwierig erweisen und die Entwickler von eingebetteten Systemen, wenn sie erstmals damit zu tun haben, vor mehrere Probleme stellen. Unwissenheit aber kann ein Entwicklungsprojekt schnell auf den Prüfstand bringen.

Entwickler müssen vor Beginn ihrer Arbeit wissen, welche Herausforderungen mit der Implementierung von USB-Wechseldatenträgern in eingebetteten Systemen verbunden sind. Sehen wir uns deshalb einen kurzen Überblick zum Universal Serial Bus (USB) und anschließend die einzelnen Schritte im Entwicklungsprozess sowie einige Überlegungen an, die diese Aufgabe so anders machen.

UNIVERSAL SERIAL BUS (USB): EIN KURZER ÜBERBLICK

- Bei USB handelt es sich um einen Standard, der Anfang der 1990er entwickelt wurde mit dem Ziel, den Anschluss von externen Geräten an PCs wesentlich zu vereinfachen. Durch seine Entwicklung und Akzeptanz gelang es, eine Vielzahl verschiedener Anschlüsse zu einem einzigen, universellen, externen Peripheriebus zusammenzuführen. Heute werden jährlich drei Milliarden USB-Geräte am Markt abgesetzt (Stand 2014, Quelle: Intel)[1]
- Die USB-Standards werden vom USB Implementers Forum (USB-IF) definiert. Entwickler, die Produkte mit integriertem USB implementieren möchten, werden ersucht, Mitglied im USB-IF zu werden. Die Mitgliedschaft ist jedoch nicht zwingend erforderlich und kostet eine jährliche Gebühr.
- Um das USB-Logo verwenden zu dürfen, muss der Hersteller die Nutzungsvereinbarung (USB Adopters Agreement) unterzeichnen und seine Geräte müssen eine Konformitätsprüfung bestehen.
- USB-Geräte müssen eine eindeutige Herstellerkennung (Vendor-ID bzw. VID) und Produktkennung (Produkt-ID bzw. PID) tragen, die sie dem Host gegenüber eindeutig identifizieren, sodass gegebenenfalls die richtigen Gerätetreiber geladen werden können. Mitgliedern des USB-IF wird die VID zugeteilt, Nichtmitglieder können sie käuflich erwerben.
- USB-Geräte enthalten Strings, die der Host abrufen kann. Diese sind aus Gründen der Internationalisierung in Unicode gespeichert, Standardsprache ist amerikanisches Englisch. Zwingend vorgeschrieben ist ein Hersteller-String, der den Namen des Geräteherstellers enthält. Weitere Strings, die z. B. den Produktnamen, die Seriennummer oder sonstige Angaben enthalten können, sind optional.
- USB existiert in verschiedenen Versionen, die sich in der Datenrate unterscheiden: USB 1.1 Low-Speed (LS) mit 1,5 Mbit/s bzw. Full-Speed (FS) mit 12 Mbit/s; USB 2.0 inklusive FS und zusätzlich Hi-Speed (HS) mit 480 Mbit/s; USB 3.0 inklusive FS und HS und zusätzlich SuperSpeed (SS) mit 5 Gbit/s; sowie USB 3.1, das die Datenrate von SS auf 10 Gbit/s erhöht. Die Datenrate gibt jeweils die Datenübertragungsrate der Busleitung an. Aufgrund von Protokoll-Overhead und Latenzzeit des Betriebssystems erreicht jedoch keine der USB-Versionen die volle Datenübertragungsrate, die auf der Leitung möglich wäre. Als Faustregel kann man sagen, dass die tatsächliche Datenübertragungsrate in der Regel max. 50–70 % der auf der Leitung möglichen Datenrate beträgt.
- Auf dem Bus sind drei verschiedene Knoten zulässig: Host, Gerät und Hub. Der Host umfasst Buchsen für den Anschluss von Hubs und Geräten und initiiert den gesamten USB-Traffic. Das Gerät wird an Hubs und Hosts angeschlossen, empfängt USB-Traffic und antwortet darauf. Beim Hub handelt es sich um ein spezielles Gerät, das einen Upstream-Port mit einem oder mehreren Downstream-Ports verbindet, den Traffic auf allen Ports wiederholt und die Antworten zurückliefert.

- USB unterstützt exakt einen Host pro Bus und bis zu 127 Geräte (einschließlich Hubs). Hubs können bis zu einem Maximum von sieben Schichten (einschließlich dem Root-Hub) verschachtelt werden.
- USB ist ein Hot-Plug-fähiger Bus, d. h. Geräte können an- und abgesteckt werden, während der Bus Strom führt. Er versorgt die angeschlossenen Geräte mit 5 V DC Nennspannung. USB 1.1 bis 2.0 liefern den Geräten bis zu 500 mA Strom, USB 3.0+ liefert bis zu 900 mA.
- USB-Geräte können auch als USB OTG (USB On-the-Go) implementiert werden. Die OTG-Spezifikation ermöglicht es dem Gerät, je nachdem, mit welchem Kabel es angeschlossen ist, entweder als USB-Host oder USB-Gerät zu funktionieren. Das ist zum Beispiel bei Digitalkameras praktisch, die an einen Computer angeschlossen als USB-Laufwerk, und an einen Drucker angeschlossen als USB-Host funktionieren sollen.
- USB-Geräte können eine Geräteklasse implementieren. Dabei handelt es sich um eine standardisierte, generische Gerätefunktion, die es Geräten ermöglicht, auf verschiedenen Betriebssystemen genau gleich zu laufen, ohne dass eigens ein Treiber installiert werden muss. Betriebssystem-Hersteller müssen für jede Klasse, die sie unterstützen, einen Klassentreiber implementieren.
- Die Geräteklasse MSC (Mass Storage Class) implementiert einen Massenspeicher und macht somit USB-Wechseldatenträger möglich.

KONZEPTIONIEREN DES SYSTEMS

Bevor Sie mit der Konzeptionierung eines eingebetteten Systems mit USB-Wechseldatenträger-Unterstützung beginnen, sollten Sie ein paar Dinge wissen und bestimmte Entscheidungen treffen, wie etwa:

Bauen oder kaufen?

Eine der ersten Überlegungen muss sein, ob Sie Ihr System selber bauen oder lieber kaufen wollen. Je größer die Wahrscheinlichkeit, dass andere die gleiche Anwendung umsetzen wollen wie Sie, desto größer die Wahrscheinlichkeit, dass es bereits etwas Entsprechendes zu kaufen gibt. Für jeden Hardware-Entwickler ist es frustrierend, im Nachhinein zu erfahren, dass bereits eine Lösung am Markt existiert und man sich all die Arbeit hätte sparen können. Nehmen Sie sich also etwas Zeit, um Single-Board-Computer und System-on-Module-Lösungen zu recherchieren und deren Preis mit den Kosten einer individuell entwickelten Hardware unter Beachtung der voraussichtlichen Stückzahlen zu vergleichen. Vergessen Sie nicht, neben den eigenen Entwicklungs- auch die Risiko-, Markteinführungs- und Opportunitätskosten zu berücksichtigen.

Konformitätsprüfung

Die USB-Konformitätsprüfung beinhaltet verschiedene, vom USB-IF vorgegebene Tests, die ermitteln, ob ein gegebenes USB-Produkt den Anforderungen der USB-Spezifikation entspricht. Sie ist zwar nicht zwingend Vorschrift, aber im Interesse der Kunden zu empfehlen. Produkte, die die Konformitätsprüfung bestanden haben, sind nicht nur garantiert mit anderen USB-Produkten kompatibel, sondern werden auch vom USB-IF als kompatibel gelistet und dürfen das USB-Logo auf ihrer Verpackung tragen.

Wie funktioniert die Konformitätsprüfung? Die Konformitätsprüfung kann entweder bei einem vom USB-IF veranstalteten Compliance-Workshop („Plugfest“) oder einem unabhängigen Testlabor erfolgen. Einen Teil der Vorabtests können Sie mithilfe von Softwaretools, die das USB-IF kostenlos zur Verfügung stellt, selbst durchführen. Diese Test-Tools stehen hier zum Download bereit [2].

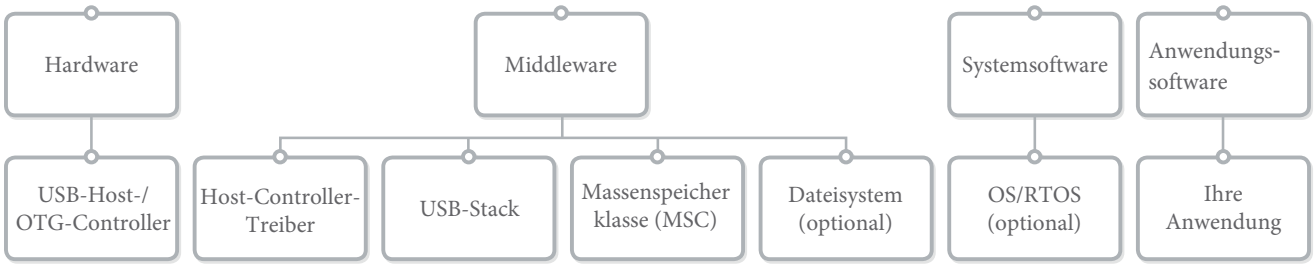
Weitere Informationen zur USB-Konformitätsprüfung finden Sie auf der Website des USB-IF [3].

Implementierung

Die USB-Spezifikation schreibt vor, dass der Lese- bzw. Schreibzugriff auf einen USB-Wechseldatenträger ausschließlich über einen USB-Host erfolgen darf. USB-Hosts sind eigens für den Anschluss anderer USB-Geräte konzipiert und besitzen zu diesem Zweck meist eine oder mehrere Steckbuchsen des Typs A. Wenn Sie also USB-Wechseldatenträger-Unterstützung in ein eingebettetes System implementieren wollen, kommen Sie nicht umhin, einen USB-Host zu implementieren.

Nicht alle eingebetteten USB-Hosts sind gleich aufgebaut. Damit ein USB-Host mit einer Geräteklasse kommunizieren kann, muss der Entwickler die Gerätetreiber, Klassentreiber und Protokollstapel für die Geräte und Dienste festlegen, die der Host unterstützen soll. Um beispielsweise einen USB-Wechseldatenträger zu unterstützen, muss der USB-Host die in **Abbildung 1** gezeigten Komponenten enthalten.

Abbildung 1: Blockschema zur Implementierung einer USB-Wechseldatenträger-Unterstützung



Wie die Abbildung oben zeigt, basiert ein eingebetteter Host für USB-Wechseldatenträger auf einer komplexen Kombination aus Hard- und Software, die als ein System zusammenwirkt.

Eine Schlüsselrolle in diesem System kommt der Software zu, die wir als „Middleware“ bezeichnen. Dahinter verbirgt sich eine Sammlung aus Treibern, Protokollstapeln und Klassentreibern, die es möglich machen, über den USB-Host-Controller mit dem USB-Wechseldatenträger zu kommunizieren. Bei eingebetteten Systemen ohne Betriebssystem kann die Middleware auch ein Dateisystem umfassen, das die auf den USB-Wechseldatenträger geschriebenen Daten so formatiert, dass andere Computer sie wie von einem Laufwerk lesen können. Auf die Middleware kommen wir später noch ausführlicher zu sprechen. Wir erwähnen sie aber bereits hier, weil man zum Verständnis der Systemebene wissen muss, dass die Middleware einen notwendigen Systembestandteil darstellt. Ebenso wichtig zu wissen ist, dass die Middleware zusätzliche Anforderungen an das System stellt, die über die der USB-Host-Controller-Hardware hinausgehen.

Kommen wir nun, da die grundlegenden Anforderungen bekannt sind, zu den verschiedenen Optionen, die sich auf Hard- und Software-Ebene bieten.

Wahlmöglichkeiten für USB-Host-Controller

Nehmen wir an, Sie haben ein individuell konzipiertes eingebettetes System, das Sie aktualisieren möchten, weil es kein USB unterstützt. Sie müssen es also um einen USB-Host-Controller erweitern. Dazu können Sie dem vorhandenen System entweder einen separaten USB-Host-Controller-Chip hinzufügen („aufpfropfen“), oder auf einen neuen Mikrocontroller mit integriertem USB-Host-Controller umsteigen.

„Aufgepfropfte“ Host-Controller

Das Hinzufügen eines „aufgepfropften“ USB-Host-Controllers, wie wir es nennen, ist mit verschiedenen Vorteilen und Kosten verbunden. Ein Argument zugunsten des alten Prozessors ist, dass auch die vorhandenen Entwicklungstools weiter verwendet werden können. Kosten fallen für den USB-Host-Controller selbst, für notwendige neue Systemressourcen wie z. B. Arbeitsspeicher (für Coderaum, den die Middleware beansprucht) und das für den Betrieb notwendige (Echtzeit-)Betriebssystem an.

USB-Host-Controller werden allgemein in drei Typen unterteilt, wie in **Tabelle 1** und nachstehend näher beschrieben:

1. **Standardisierte USB-Host-Controller:** Diese Host-Controller implementieren eine Standard-Softwareschnittstelle und erfordern in der Regel ein Betriebssystem oder Echtzeitbetriebssystem sowie Middleware, die zusätzlichen Arbeits- und Flash-Speicher beansprucht.

2. **Proprietäre USB-Host-Controller:** Diese Host-Controller implementieren eine herstellerspezifische Softwareschnittstelle und setzen die Nutzung vom Hersteller bereitgestellter oder selbsterstellter Bibliotheken voraus. Auch sie können Middleware erfordern, die allerdings den benötigten Chip unterstützen muss, da die Schnittstelle herstellergebunden ist.

3. **Peripherer USB-Host:** Diese Chips sind genau genommen eigenständige eingebettete Systeme, die einen USB-Host-Controller sowie Firmware inklusive Middleware, Massenspeicherklasse und Dateisystem in einem einzigen Chip bündeln. Sie sind einfach im Gebrauch und bieten bei minimalem Platzanspruch zwar nur eingeschränkte Funktionen und Leistung, die aber für eine Vielzahl von Anwendungen durchaus ausreichen.

Tabelle 1: Auszug aus dem Angebot an USB-Host-Controller-Chips

Hersteller	Typ	Artikelnummer(n)	Bus	Quellen
NXP	Standardisiert	SAF1562HL	PCI	NXP
	Standardisiert	SAF1760BE	Generisch	Schnittstellenprodukte [4]
	Standardisiert	SAF1761BE	Generisch	
Cypress	Proprietär	SL811HS	Generisch	
	Proprietär	CYC67200	Generisch	Cypress USB-Hosts [5]
	Proprietär	CYC67300	Generisch	
FTDI	Peripher	VNC1L	Parallel,,	
	Peripher	VNC2	UART, SPI	FTDI-Produkte [6]
	Proprietär	FT313H	Parallel	

Ein entscheidender Aspekt bei der Wahl des Chip-Typs ist, ob Ihr System über ausreichend Ressourcen für ein Betriebssystem bzw. Echtzeitbetriebssystem plus Middleware verfügt. Wenn Ihr System eine hohe Datendurchsatz-Leistung wie z. B. Hi-Speed oder SuperSpeed verlangt und ein OS bzw. RTOS sowie Middleware problemlos unterstützen kann, ist ein standardisierter Host-Controller wahrscheinlich eine gute Wahl. Bei Systemen, die eine höhere Leistung erfordern und zwar Middleware, aber kein OS oder RTOS unterstützen können, passt vermutlich eher ein proprietärer Host-Controller. Wenn Ihr System nur über eingeschränkte Ressourcen verfügt und eine moderate Datendurchsatz-Leistung hinnehmbar ist, ist ein gebrauchsfertiger peripherer USB-Host wahrscheinlich die bessere Wahl, da er das eingebettete System weniger beansprucht.

Integrierte USB-Host-Controller

Wenn Sie die Lösung mit einem „aufgepfropften" Host-Controller nicht anspricht, können Sie Ihr System alternativ mit einem Prozessor konzipieren, der einen integrierten USB-Host-Controller besitzt. Normalerweise käme dieser Weg für das Upgrade eines bestehenden Systems nicht infrage, da es im Grunde bedeutet, das komplette System auszutauschen. In den letzten Jahren wurden allerdings viele billige, verbrauchsarme und leistungsstarke Mikrocontroller auf den Markt gespült. Diese neuen Chips enthalten eine Reihe hochwertiger Peripheriefunktionen, viele von ihnen mit integrierter USB-Host-Unterstützung inklusive kostenloser Middleware und RTOS-Unterstützung, die in der Design-Software für den Chip enthalten ist. Solche herstellerseitig bereitgestellte Firmware ist von Grund auf für den jeweiligen Chip konzipiert, was seine Integration vereinfacht. Ein weiterer Vorteil ist, dass nicht nur Sie, sondern Hunderte anderer mit dem gleichen Chip arbeiten. Damit steigt die Wahrscheinlichkeit, sich in der Entwicklergemeinde über das System und die Tools austauschen und kostenlose Tipps und Hilfe bekommen zu können. **Tabelle 2** zeigt einen Bruchteil der vielen Mikrocontroller, die über integriertes USB OTG bzw. Host-Support verfügen.

Tabelle 2: Auszug aus dem Angebot an Mikrocontrollern mit integriertem Host-Support

Hersteller	Serie	Architektur	USB OTG oder Host-Support	Quellen
NTI	Tiva	ARM	FS/HS-Gerät u. OTG	Tiva C-Serie MCU Website [7]
TI	Hercules	ARM	LS/FS OHCI, FS-Gerät	TI Safety Mikrocontroller [8]
TI	Sitara	ARM	FS/HS-Host u. Gerät	TI Sitara Mikroprozessoren [9]
TI	OMAP	ARM	FS/HS-Host u. Gerät	TI OMAP Anwendungsprozessoren [10]
Freescale	Kinetis	ARM	FS/HS-Host u. Gerät	Freescale Kinetis Mikrocontroller [11]
Freescale	Vybrid	ARM	FS/HS-Host u. Gerät	Freescale Vybrid Mikroprozessoren [12]
Freescale	i.MX	ARM	FS/HS-Host u. Gerät	Freescale i.MX-Serie Mikroprozessoren [13]
Atmel	AVR	AVR	FS OTG u. Gerät	Atmel AVR Mikrocontroller [14]
Atmel	SAM	ARM	FS-Host u. Gerät	Atmel ARM Mikrocontroller [15]
NXP	LPC	ARM	FS/HS-Host u. Gerät	NXP LPC Mikrocontroller [16]
Microchip	PIC	PIC	FS/HS-Host u. Gerät	Microchip PIC Mikrocontroller [17]

USB-Software-Unterstützung

Betriebssystem

Eine Möglichkeit, um die USB-Software-Unterstützung in Ihrem System sicherzustellen, besteht in der Verwendung eines Betriebssystems. Aus Software-Sicht bieten Betriebssysteme eine attraktive Lösung, da sie Gerätetreiber für die meisten Host-Controller, den USB-Stack, einen Klassentreiber für Massenspeicher und ein Dateisystem beinhalten. Darüber hinaus gehören auch extrem leistungsstarke Entwicklungs-Tools in ihren Lieferumfang. Der Nachteil allerdings ist, dass ein Betriebssystem erhebliche Systemressourcen und viel Flash- und Arbeitsspeicher beansprucht und relativ langsam bootet. Bei kommerziellen Betriebssystemen kann pro Stück eine Lizenzgebühr anfallen, was bei Open-Source-Betriebssystemen in der Regel nicht der Fall ist. **Tabelle 3** zeigt einige der eingebetteten Betriebssysteme mit integrierter USB-Wechseldatenträger-Unterstützung.

Tabelle 3: Auszug aus dem Angebot an eingebetteten Betriebssystemen

Name	Hersteller	Architektur	Lizenz	Quellen
Linux	N. z.	Viele	GNU	www.linux.org
Windows Embedded	Microsoft	x86	Windows EULA	Windows Embedded [18]
NetBSD	N. z.	Viele	Berkeley	www.netbsd.org

Echtzeitbetriebssysteme und Middleware von Drittanbietern

Einige Anwendungen erfordern den Einsatz eines Echtzeitbetriebssystems (RTOS), um den Anforderungen in Bezug auf Determinismus, Durchsatz und/oder Antwortzeit zu genügen. Die meisten RTOS-Produkte bieten USB-Unterstützung in Form von Middleware, die in einer Entwicklungs-Tool-Suite enthalten ist. Leider sind diese Middleware-Produkte meist proprietär, mit den Tools anderer Hersteller üblicherweise nicht kompatibel, und teuer – bis zu einigen Tausend Dollar pro Lizenz. **Tabelle 4** zeigt einige gängige kommerzielle RTOS-Lösungen.

Tabelle 4: Auszug aus dem Angebot an RTOS- und Middleware-Lösungen von Drittanbietern

Name	Hersteller	Komponenten	Quellen
emWare	Segger	embOS, emboOS/IP, emWin, emFile, emUSB-Gerät, emUSB-Host, emLib	emUSB Embedded Host [19]
MDK	Keil	RTX, CAN, Flash-Dateisystem, USB-Host, USB-Gerät, TCP/IP, GUI	Keil ARM Tools [20]
INTEGRITY	Green Hills	Networking, Kryptographie, Dateisystem, USB, Grafik	Green Hills Tools [21]
MQX	Freescale	RTOS, RTCS, MFS, USB-Host/Gerät	Freescale MQX Tools [22]
uC/OS	Micrium	RTOS, TCP/IP, USB-Host/Gerät, CAN, Modbus, Bluetooth, Dateisystem, GUI	www.micrium.com

Echtzeitbetriebssysteme und Middleware vom Hersteller

Wenn die Lösung mit einem „integrierten“ USB-Host-Controller für Sie infrage kommt, haben Sie vielleicht die Chance, das Echtzeitbetriebssystem und die Middleware vergünstigt zu erhalten. Wie bereits erwähnt sind die Hersteller von Mikrocontrollern daran interessiert, Sie als Kunden für ihre neuen Produkte zu gewinnen. Manche legen deshalb RTOS und Middleware kostenlos oben drauf, um Ihnen die Entscheidung zu erleichtern. **Tabelle 5** zeigt, welche RTOS- und Middleware-Produkte für verschiedene Prozessoren angeboten werden.

Tabelle 5: Auszug aus dem Angebot an RTOS- und Middleware-Lösungen von Mikrocontroller-Herstellern

Hersteller	Serie	Architektur	Quellen
TI	Tiva	ARM	TivaWare Hardware-/Software-Ressourcen Website [23]
Atmel	AVR	AVR/AVR32	ASF USB-Host-Stack Anwendungshinweis [24]
Freescale	Alle	Alle	MQX Software-Lösungen [25]

Bare Metal

Wenn Sie einen peripheren USB-Host für Ihre Host-Controller-Lösung verwenden, haben Sie die Möglichkeit, ein sogenanntes „Bare Metal“ eingebettetes System zu implementieren. Bare-Metal-Systeme besitzen weder ein Betriebssystem noch ein Echtzeitbetriebssystem, sondern sind lediglich ein Programm, das auf einem Mikrocontroller läuft. Moderne Bare-Metal-Systeme sind üblicherweise in C geschrieben und senden die Befehle an den peripheren USB-Host direkt über standardübliche Peripheriegeräte-Schnittstellen. Bare-Metal-Systeme unterstützen aufgrund ihrer minimalistischen Implementierung meist nur die absoluten Basics, die von ihnen implementierten Funktionen sind in der Regel aber extrem effizient, weil sie weder im Coderaum noch im Arbeitsspeicher oder den Systemressourcen Overhead für das Betriebssystem bzw. Echtzeitbetriebssystem beanspruchen.

KONZEPTIONIEREN INDIVIDUELLER USB-HARDWARE

Wenn Sie sich dafür entscheiden, Ihre USB-Hardware selbst maßzuschneidern, gilt es einige grundlegende Dinge zu beachten. Hier ein paar hilfreiche Tipps, die Sie berücksichtigen sollten:

USB-Anschlüsse

USB-Wechseldatenträger für den Endanwenderbereich besitzen einen Stecker des Typs A und werden in eine Buchse des Typs A eingesteckt. Die Belegung dieser Buchsen ist in **Abbildung 2** gezeigt. Bei den USB-Buchsen des Typs A handelt es sich um eine kostengünstige Variante für den nicht-gewerblichen Bereich, die auf mindestens 1.500 Steckzyklen ausgelegt ist. Die USB-Spezifikation enthält zwar auch Spezifikationen für den Anschluss, diese sind allerdings nicht verbindlich. Für spezielle Anwendungen, die z. B. eine hohe Zuverlässigkeit voraussetzen, können die gängigen Anschlüsse unter Umständen ungenügend sein. Standardübliche USB-Anschlüsse des Typs A sind beispielsweise nicht wasserdicht, weshalb Drittanbieter spezielle Ausführungen entwickelt haben, die den Anforderungen ihrer Kunden gerecht werden. **Abbildung 3** zeigt eine Einbaubuchse, die in eine herkömmliche USB-Buchse des Typs A eingesteckt werden kann, aber eine extrem hohe Lebensdauer bietet und in IP65- sowie IP67-tauglicher Ausführung erhältlich ist. Achten Sie bei der Festlegung der Anforderungen Ihres eingebetteten Systems unbedingt darauf, den richtigen Anschluss zu spezifizieren, der Ihrem Bedarf entspricht.

Abbildung 2: Pinbelegung einer USB-Buchse des Typs A

Pin	Name
1	VBUS
2	D-
3	D+
4	GND

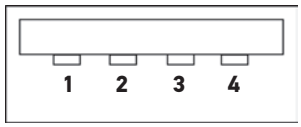


Abbildung 3: Buchse und RUGGEDrive™ Speicherstick der UR-Serie von Datakey

Abbildung 3 zeigt eine Buchse der UR-Serie von Datakey, die in einen herkömmlichen USB-Port eingesteckt, aber zusammen mit den RUGGEDrive™-Speichersticks von Datakey genutzt wird. Diese Buchsen bieten eine hohe Lebensdauer von 50.000 Steckzyklen (gegenüber nur 1.500 bei standardüblichen USB-Buchsen des Typs A) und werden in verschiedenen IP-tauglichen Ausführungen angeboten.

Stromversorgung

Die wichtigste Schaltung in jeder Elektronik ist die Stromversorgung: So wie der Mensch saubere Luft zum Atmen braucht, brauchen elektronische Geräte sauberen Strom, um funktionieren zu können. Einer gut funktionierenden Stromversorgung schenkt kaum jemand Beachtung. Sobald aber die Stromversorgung nicht funktioniert, kann selbst das beste System nicht laufen. Bei eingebetteten USB-Hosts ist die Stromversorgung doppelt wichtig, weil sie nicht nur den Host selbst, sondern auch die an ihn angeschlossenen Geräte mit Strom versorgt.

USB-Hosts müssen auf der VBUS-Leitung geregelten, gegen Überstrom gesicherten Strom mit einer Spannung von 4,75 bis 5,0 V bereitstellen. Die Stromstärke wird in „Ladungseinheiten“ angegeben, wobei eine Ladungseinheit 100 mA entspricht. Extern gespeiste Full- und Hi-Speed-Hosts müssen bis zu fünf, SuperSpeed-Hosts bis zu neun Ladungseinheiten Strom für die angeschlossenen Geräte bereitstellen. Akkugespeiste Hosts können mitunter nur eine einzige Ladungseinheit bereitstellen.

Strombedarf

Ein USB-Gerät sollte beim ersten Anstecken an ein System nicht mehr als eine Ladungseinheit (100 mA) Strom ziehen.

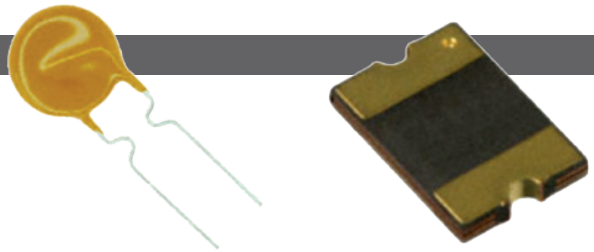
Der Host fragt den erforderlichen Betriebsstrom vom Gerät ab, das Gerät antwortet ihm mit dem voraussichtlichen Stromverbrauch. Wenn der Host nur eine begrenzte Strommenge liefern kann, kann er dem Gerät die Stromversorgung verweigern, indem er schlicht keine Konfigurationsmeldung sendet. Empfängt das Gerät keine Konfigurationsmeldung, kommt keine Verbindung zustande. Für den Anwender ist es nicht unbedingt ersichtlich, dass oder warum die Verbindung fehlgeschlagen ist. Aus diesem Grund sollte der USB-Host den Anwender benachrichtigen, wenn er die Stromversorgung eines Geräts verweigert, sodass der Anwender weiß, was passiert ist, und gegebenenfalls Abhilfemaßnahmen ergreifen kann.

Bei extern gespeisten eingebetteten USB-Hosts ist es üblich, VBUS über eine Überstromschutzeinrichtung an die +5 V-Schiene anzuschließen und alle Anfragen um Stromversorgung zu bestätigen. Wenn Ihr System dagegen über Akku gespeist wird oder sehr strenge Anforderungen an den Stromverbrauch stellt, sollten Sie es auf ein rigoroseres Strommanagement auslegen. Ausführlichere Informationen über das USB-Strommanagement sind den Abschnitten 7.2 und 9.2.5 der USB 2.0 Spezifikation [26], sowie Abschnitt 2.1.1 des USB 3.0 On-The-Go and Embedded Host Supplement [27] zu entnehmen.

Überstromschutz

Die USB-Spezifikation schreibt vor, dass jeder USB-Host, d. h. auch eingebettete Hosts, gegen Überstrom gesichert sein muss. Diese Vorschrift besteht aus Sicherheitsgründen für den Fall, dass ein in den Host eingestecktes Gerät oder Kabel die stromführende Leitung (VBUS) zur Masse (GND) kurzschließt. Ohne Strombegrenzung könnte in diesem Fall ein Brand entstehen, wenn die Stromquelle ausreichend stark ist.

Abbildung 4: Rückstellbare PTC-Sicherung



Die Anforderung besagt, dass die Überstrombegrenzung ohne menschliches Eingreifen rückstellbar sein muss, was eine Sicherung ausschließt. PC-Hosts enthalten üblicherweise einen Root-Hub-Chip, der den Strom erfassen kann, sowie externe VBUS-Schalter, die die Stromversorgung zu Geräten, die zu viel Strom ziehen, abschalten. Das ist auch bei eingebetteten Systemen möglich. Um jedoch die Kosten niedrig zu halten, enthalten die meisten eingebetteten Hosts eine rückstellbare PTC-Sicherung (wie z. B. die in **Abbildung 4** gezeigten) zwischen Stromquelle und VBUS-Leitung. Bei einer PTC-Sicherung (auch unter Namen wie „Polyfuse“ oder „Polyswitch“ bekannt) handelt es sich um eine selbstrückstellende Sicherung auf Basis eines Festkörpermaterials, das hochohmig wird, wenn der Strom einen bestimmten Wert übersteigt, und von selbst zurück in einen niederohmigen Zustand wechselt, wenn der Überstrom behoben ist.

Einschaltstromschutz

Wenn ein Gerät an einen USB-Host angeschlossen wird, muss der Host den Bypass-Kondensator im Gerät laden und gleichzeitig das Gerät mit Strom versorgen. Dadurch kommt es zu hohen Einschaltströmen, die, wenn sie nicht begrenzt werden, zu einem Spannungsabfall in der Stromversorgung und in der Folge zu instabilem Systembetrieb führen können.

Eine Möglichkeit für USB-Hosts, hohe Einschaltströme zu bewältigen besteht darin, auf der Host-Seite des VBUS einen größeren Stützkondensator mit einer Kapazität von rund 120 µF vorzusehen, der als Ladungsspeicher dient und beim Einstecken von Geräten Strom liefert, der sonst von der Stromversorgung des Hosts gezogen würde und diese überlasten könnte. Eine zweite Abhilfemaßnahme, die in Kombination mit einem größeren Stützkondensator eingesetzt werden kann besteht darin, zwischen Stützkondensator auf Host- und VBUS auf Geräteseite einen kleinen Reihenwiderstand anzuordnen. Dadurch wird der maximale Einschaltstrom, der einem Gerät bereitgestellt werden kann, begrenzt. Zusätzlich entsteht durch den Reihenwiderstand aber auch ein konstanter Spannungsabfall im Normalbetrieb, der allerdings vernachlässigt werden kann, solange die Ausgangsspannung am Host-Port zwischen 4,75 und 5,0 V beträgt.

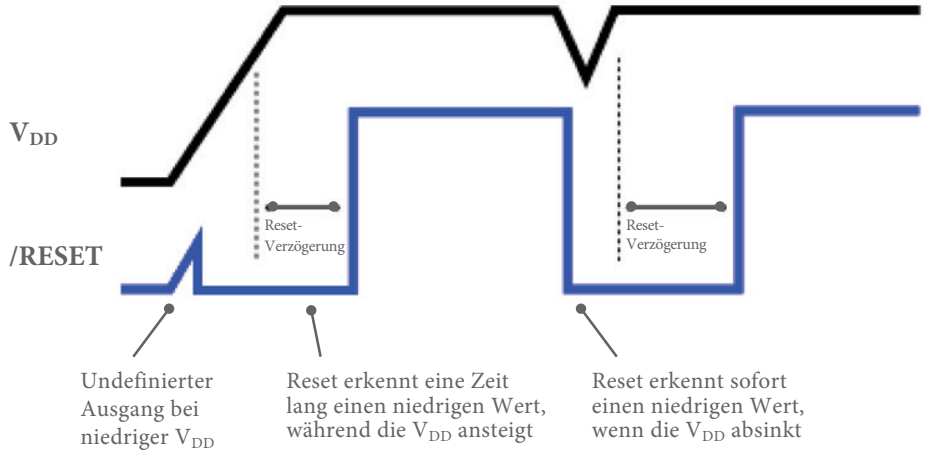
Reset-Kontrolle und Brownout-Erkennung

Jedes mikrocontroller-basierte System sollte eine integrierte Reset-Kontrolle und Brownout-Erkennung besitzen. Die Reset-Kontrolle kann relativ einfach ausgeführt sein, zum Beispiel als RC-Schaltung zwischen Phase und Masse, die den Mikrocontroller sanft aus dem Reset holt, sobald die Stromversorgung stabil ist.

Wussten Sie, dass ein Mikrocontroller ohne Reset-Controller selbst nach Abschalten des Stroms weiterläuft? Das ist tatsächlich so! Nach dem Abschalten fährt die Versorgungsspannung langsam auf Null herunter. In dieser Zeit ist die Spannung an den Stromanschluss-Pins des Mikrokontrollers immer noch hoch genug, um ihn am Laufen zu halten. Manche Teile des Mikrocontrollers brauchen sehr wenig Strom und laufen lange Zeit weiter, während andere mit höherem Strombedarf relativ schnell abschalten. Dieses Phänomen ist als „Brownout" bekannt und bezeichnet den Zustand beim Hoch- bzw. Herunterfahren der Stromversorgung, in dem einzelne Teile des Systems schon (oder noch) laufen, und andere nicht. Bei flash-basierten Mikrocontrollern kann der Brownout zu einem massiven Problem führen, weil der Mikrocontroller in diesem Zustand in der Lage ist, zufälligen Code auszuführen, der den Flash-Speicher korrumpieren kann. **Abbildung 5** zeigt, wie ein Überwachungselement den Mikrocontroller schützt, indem es ihn zurückstellt, sobald die Eingangsspannung zu niedrig wird.

Abbildung 5: Brownout infolge Störimpuls [28]

So schützt ein Überwachungselement vor Brownout



Eine Lösung zum Schutz vor Brownout besteht in einem Überwachungselement bzw. Reset-Controller. Dabei handelt es sich um eine einfache integrierte Schaltung, die die Rückstellung nicht anhand der Zeit, sondern anhand der Versorgungsspannung steuert. Reset-Controller, die dem Brownout ein Ende bereiten, werden von verschiedenen Herstellern angeboten. Das Diagramm in **Abbildung 5** zeigt, wie Überwachungs- und Reset-ICs von Texas Instruments funktionieren. Denken Sie daran, einen solchen Schutz auch in Ihr System einzuplanen.

Elektromagnetische Verträglichkeit (EMV)

Die Anforderungen an elektromagnetische Störausstrahlung und Störfestigkeit von elektronischen Produkten sind in verschiedenen Vorschriften (FCC, CE usw.) geregelt. Eine Störausstrahlung kann von den Datensignalen wie auch vom Masseanschluss des Systems ausgehen. Hier ein paar schnelle Tipps, wie sich die Störausstrahlung minimieren lässt:

1. Setzen Sie auf Ihren PCBs Flächen ein: Sie bieten den induktivitätsärmsten Pfad für Spannung und Masse und verringern die Störausstrahlung.

2. Sehen Sie auf der obersten und untersten Schicht geerdete polygonförmige Leiterbahnen als Rückweg für asymmetrische Rückströme vor.

3. Überbrücken Sie alle ICs mit Keramikkondensatoren mit niedrigem ESR.

4. Setzen Sie an den D+/D-Leitungen der USB-Anschlüsse einen EMV-Filter ein, der auf die Datenübertragungsrate des Anschlusses abgestimmt sein muss.

5. Setzen Sie an den VBUS- und GND-Leitungen Ferritperlen ein, um vom System erzeugtes Rauschen auszufiltern, sodass es nicht auf die Adern des USB-Kabels gelangt.

6. Schließen Sie die USB-Schirmung nicht direkt an die Masse an, sondern über einen 1 MΩ-Widerstand, der mit einem 4.700 µF-Keramikkondensator parallel geschaltet ist. Dies sollte eine ausreichende Filterwirkung bieten, um Rauschstrahlung durch die Schirmung zu verhindern.

7. Verwenden Sie ein Gehäuse aus Metall oder metallisiertem Kunststoff als faradayschen Käfig, um von Ihrer Elektronik ausgehende Störstrahlung zurückzuhalten.

8. Schließen Sie die Schirmanschlüsse am USB-Anschluss elektrisch an das Metall- bzw. metallisierte Kunststoffgehäuse an, um die Störausstrahlung weiter zu verringern.

Impedanzkontrolle

USB-Signale sind differenzielle Signale und müssen über ein Layout mit einer differenziellen Impedanz von 90 Ω +/-10 % geführt werden. Was kontrolliert werden muss, sind die Signale auf den D+ und D- Leitungen zwischen Anschluss und Host-Controller. Bei Low-Speed- und Full-Speed-Designs ist eine gut ausgeführte Impedanzkontrolle eine gute Sache, bei Hi-Speed- und SuperSpeed-Designs aber ist sie absolut unverzichtbar. Denn ohne sie würden die Schaltungen schlicht nicht funktionieren. Falls Sie eine Konformitätsprüfung vornehmen lassen, wird auch die Impedanz Ihres Systems im TDR (Zeitbereichsreflektometer) gemessen um sicherzustellen, dass sie korrekt ausgelegt ist.

Ein impedanzkontrolliertes Layout ist an und für sich nicht schwierig. Das Grundprinzip besteht darin, die D- und D + Signale über die gesamte Strecke in einem festen Abstand nebeneinander über eine durchgängige Massefläche zu führen. Versuchen Sie, D+ und D- gleich lang und auf der gleichen Schicht zu halten. Als Routingtopologie kommen Microstrip (Streifenleitung) und Stripline (symmetrische Streifenleitung) infrage, wie in **Abbildung 6** gezeigt.

Abbildung 6: PCB-Layout mit differenziellen Paaren

Cypress Semiconductor hat einen hervorragenden Anwendungshinweis für Hi-Speed USB-Layouts herausgebracht, der sich definitiv zu lesen lohnt29. Ein weiteres praktisches Tool, um Impedanzkontrolle im PCB-Layout zu integrieren, bieten spezielle Impedanz-Berechnungsprogramme wie z. B. das PCB Toolkit von Saturn PCB Design [30].

Eine andere Möglichkeit besteht darin, dem Platinenhersteller die impedanzkontrollierten Signale in der Fertigungsanweisung zu spezifizieren und die Leiterplatte mit der richtigen Impedanzkontrolle zwischen den USB D +/D- Bahnen fertigen zu lassen. Die meisten Platinenhersteller verfügen über die notwendige Software, um die Impedanz exakt zu berechnen, und sind in der Lage, die Impedanz der Leiterbahnen im TDR zu messen. Diese Praxis ist recht gängig und kein großer zusätzlicher Kostenfaktor, und somit ein erstklassiger Weg, um sich impedanzkontrollierte Leiterplatten mit minimalem Risiko zu beschaffen. Legen Sie das Layout der differenziellen Leiterbahnen mithilfe des Anwendungshinweises und der Ergebnisse des Impedanzrechners fest, und fügen Sie dann in den Schaltplan Anweisungen für den Platinenhersteller ein, die Leiterplatte mit 90 Ohm differenzieller Impedanz zwischen den USB-Bahnen zu fertigen. Kennzeichnen Sie die betreffenden Bahnen im Schaltplan ganz genau, sodass der Hersteller weiß, um welche es geht.

Fazit: Sie können sich die Impedanzkontrolle auf der Leiterplatte erheblich vereinfachen, indem Sie auf einen Impedanzrechner und/oder die Hilfe eines Platinenherstellers zurückgreifen.

Dimensionierung der Leiterbahnbreite

Bei den Signalbahnen ist die Bahnbreite auf der Leiterplatte im Allgemeinen durch die Mindestgeometrie für den Platinenprozess beschränkt. Bei den Strombahnen dagegen kann sich die Bahnbreite erheblich darauf auswirken, wie gut die Schaltung funktioniert. Wie viel Strom eine Bahn auf der Leiterplatte führen kann, hängt davon ab, welcher Temperaturanstieg hinnehmbar ist. Der Temperaturanstieg wird von der Dicke der Kupferbahn und ihrer Breite bestimmt. Eine einfache Möglichkeit zur Dimensionierung der Leiterbahnen bieten Rechner wie dieser [31]. Die Breite der Strombahnen auf der Leiterplatte zu berechnen, mag übertrieben wirken. Aber der Aufwand ist allemal besser, als später eine unliebsame Überraschung zu erleben.

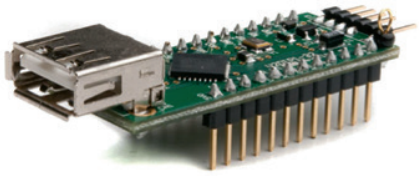
HOST-CONTROLLER: ZWEI BEISPIELLÖSUNGEN

Nun wollen wir uns zwei verschiedene USB-Host-Controller-Lösungen einmal etwas genauer ansehen. Das erste Beispiel beschreibt einen „aufgepfropften“ Host-Controller, das zweite einen Mikrocontroller mit integriertem USB-Host-Controller.

Beispiellösung für einen aufgepfropften Host-Controller (Bolt-On Host Controller)

Nehmen wir für dieses Beispiel den FTDI V2DIP1. Dieses Modul ist etwas kleiner als ein Streifen Kaugummi, aber enthält alle notwendigen Schaltungen für einen USB-Host-Controller einschließlich Anschluss, wie in **Abbildung 7** zu sehen.

Abbildung 7: V2DIP1 USB-Host-Modul



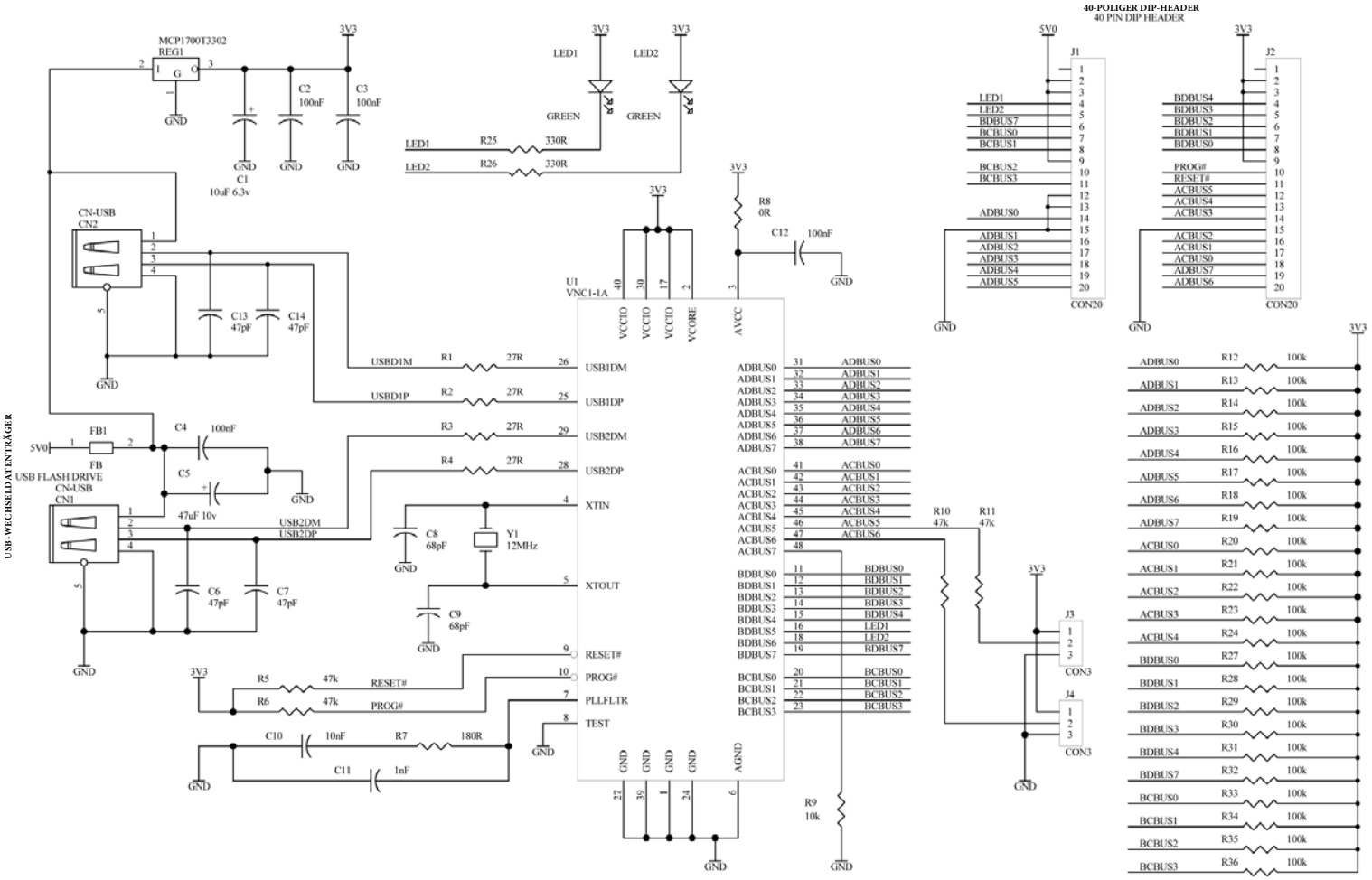
Das Modul basiert auf dem Vinculum II, einem Full-Speed USB-Host-SoC (System-on-Chip) der zweiten Generation, der den Host-Controller, die Middleware und die FAT-Dateisystem-Firmware auf einem einzigen Chip enthält.

Aus Hardware-Sicht können Sie entweder das Modul oder den Chip in Ihre Schaltung einbinden. In beiden Fällen müssen Sie eine Verbindung zur Stromversorgung herstellen und die UART-, parallele FIFO- oder SPI-Schnittstelle mit dem Mikroprozessor verdrahten. Das Modul enthält zwar nur einen USB-Host-Anschluss, unterstützt aber tatsächlich zwei USB-Host-Ports. Sie müssen lediglich den zweiten Port verdrahten, wenn Sie ihn nutzen wollen. **Abbildung 8** zeigt das Schaltbild des V2DIP1.

Die USB-Host-Controller-Funktionen werden ausgeführt, indem der Mikrocontroller über die Mikrocontroller-Schnittstelle High-Level-Befehle an den Vinculum II sendet. Um zum Beispiel das Verzeichnis eines USB-Wechseldatenträgers auszulesen, wird der Befehl „DIR“ über die Mikrocontroller-Schnittstelle gesendet. Das Modul gibt in Antwort darauf eine Liste der Dateinamen zurück. Das Befehlsprotokoll unterstützt entweder ASCII- oder binäre Befehle.

Alles in allem ist ein peripherer USB-Host eine schlaue Lösung für Anwendungen, bei denen die Ressourcen begrenzt sind und eine vergleichsweise geringe Datenrate ausreicht.

Abbildung 8: Schaltbild des V2DIP1



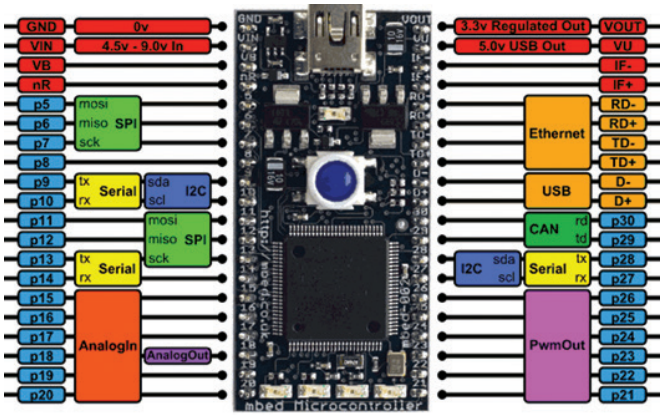
Abdruck mit freundlicher Genehmigung von FTDI.

Beispiellösung für einen integrierten Host-Controller (Built-In Host Controller)

Für dieses Beispiel nehmen wir die Entwicklungsplatine NXP LPC1768 von mbed. Dabei handelt es sich um einen winzigen Einplatinencomputer auf Basis der NXP LPC1768 ARM Cortex-M3 MCU. **Abbildung 9** zeigt das Modul in der Draufsicht inklusive Beschreibung der einzelnen Pins. Die Hard- und Software unterliegt der Apache 2.0-Lizenz, die die lizenzgebührenfreie Nutzung in privaten und kommerziellen Produkten gestattet.

Bei dem Mini-USB-Port am Modul handelt es sich um einen Geräteport, der für Software-Downloads und Debugger-Funktionen an einen PC angeschlossen wird. Um dieses Board als USB-Host-Controller zu verwenden, müssen Sie den USB-Host-Anschluss mit den im mbed HDK [32] genannten Ports verdrahten (**Abbildung 10**).

Abbildung 9: NXP LPC1768 Entwicklungsplatine von mbed

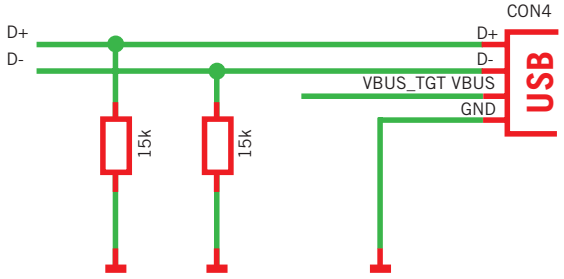


Abdruck mit freundlicher Genehmigung von mbed.

Die Software-Entwicklungstools für mbed stehen kostenlos im Internet zur Verfügung und machen das Entwickeln von Code so einfach wie das Erstellen eines Web-Accounts. Auch alle IDE-Compiler, Versionsverwaltungssysteme und Bibliotheken sind im Internet verfügbar. Das System ist darauf ausgerichtet, Quellcode, den andere Entwickler der Gemeinde bereitstellen, ganz einfach importieren zu können. Ein Entwickler beispielsweise hat ein Programm namens „MSCUsbHost" entworfen und veröffentlicht, das hier heruntergeladen werden kann [33]. Dieses Programm nutzt die enthaltenen Bibliotheken, Middleware und FAT-Dateisystem, um das Verzeichnis eines USB-Wechseldatenträgers auszulesen, eine Testdatei zu schreiben und den Inhalt dieser Testdatei zurückzulesen. Sie können diesen Code einfach in Ihr Online-Konto importieren und sofort loslegen.

Um sich den Aufbau Ihrer eigenen Hardware mit dem NXP LPC1768 zu vereinfachen, können Sie den Schaltplan der mbed NXP LPC1768 Entwicklungsplatine unter www.mbed.org herunterladen.

Abbildung 10: Verdrahtung des USB-Host-Anschlusses mit dem NXP LPC1768 von mbed



IMPLEMENTIEREN DER EINGEBETTETEN FIRMWARE

Nachdem die Hardware gekauft oder gebaut und die Systemsoftware installiert ist, ist es Zeit, Ihre Anwendung zu schreiben bzw. anzupassen. Hier ein paar Dinge, die Sie dabei beachten sollten:

Dateisystem des USB-Wechseldatenträgers

Die meisten USB-Wechseldatenträger sind werksseitig im FAT32- oder ExFAT-Dateisystem vorformatiert. Das heißt aller Wahrscheinlichkeit nach werden die USB-Wechseldatenträger, die später an Ihr System angeschlossen werden, in einem dieser Dateisysteme formatiert sein. ExFAT ist nicht mit dem FAT-Dateisystem kompatibel. Wenn Ihr System also nur FAT unterstützt, muss es in der Lage sein zu erkennen, ob ein neu angeschlossener USB-Wechseldatenträger in FAT formatiert ist oder nicht, und gegebenenfalls dazu auffordern, ihn neu zu formatieren.

Unerwartetes Ausstecken und Stromausfall

So praktisch USB-Wechseldatenträger auch sein mögen, bringen sie doch eine Unwägbarkeit mit sich, die bei der Entwicklung von eingebetteten Systemen bedacht werden muss: Nämlich die simple Tatsache, dass ein USB-Wechseldatenträger ausgesteckt werden kann, während ein Lese- oder Schreibvorgang läuft. Bei eingebetteten Systemen stellen zudem auch plötzliche Stromausfälle ein Problem dar.

Das FAT-Dateisystem mindert die Folgen eines unerwarteten Aussteckens oder Stromausfalls, indem es zwei Kopien der Dateizuordnungstabelle (File Allocation Table bzw. FAT) führt. Wird der Wechseldatenträger ausgesteckt oder tritt ein Stromausfall auf, während gerade die Tabelle geschrieben wird, ist die Wahrscheinlichkeit hoch, dass nur eine der beiden Tabellen beschädigt ist und die Daten wiederhergestellt werden können. ExFAT führt nur eine Kopie der Dateizuordnungstabelle, weshalb die Wiederherstellbarkeit bei diesem Dateisystem weniger wahrscheinlich ist.

Eine weitere Möglichkeit, der Korrumpierung von Datenträgern bei plötzlichem Ausstecken oder Stromausfall vorzubeugen besteht darin, die Software so auszulegen, dass Dateizugriffe in möglichst kurzer Zeit abgeschlossen werden. Statt eine Datei zu Beginn der Programmausführung zu öffnen und lange Zeit offen zu lassen, ist es besser, die Datei erst dann zu öffnen, wenn Daten gelesen oder geschrieben werden sollen, und anschließend sofort wieder zu schließen. Auf diese Weise wird die Gefahr der Dateikorrumpierung minimiert.

Wenn Ihr Dateisystem ein Caching-Schema implementiert, ist es am besten, das Schreib-Caching zu deaktivieren, da es aufgrund von asynchronem Cache-Flushing die Dauer des Schreibvorgangs verlängern kann.

Lizenzierung des Dateisystems

Das FAT-Dateisystem ist eines der beliebtesten Dateisysteme unserer Tage. Sowohl FAT als auch ExFAT sind durch Softwarepatente geschützt, deren Rechte bei Microsoft liegen. Wenn Sie bei Ihrem Produkt ein FAT-Dateisystem nutzen wollen, müssen Sie bei Microsoft nachfragen, ob in Ihrem Fall eine Lizenz benötigt wird. Wenn Sie ExFAT nutzen wollen, müssen Sie auf jeden Fall eine Lizenz erwerben. Ausführlichere Informationen sind auf der Website zu den Lizenzprogrammen von Microsoft zu finden.

ÜBER DATAKEY

Die Marke Datakey von ATEK Access Technologies hat sich seit 1976 einen Namen für robuste mobile Speicherlösungen für den Bereich der eingebetteten Elektronik geschaffen. Die Datakey Produktserie RUGGEDrive™ bietet verschiedene USB-Speichersticks, die vielen Schwachpunkten handelsüblicher Produkte ein Ende setzen, darunter USB-Sticks in extrem robuster Ausführung, mit erweitertem Betriebstemperaturbereich, Schreibschutz, weiteren Sicherheitsmerkmalen sowie OEM-individualisierbaren Merkmalen. Darüber hinaus sind in dieser Produktserie auch IP65- und IP67-taugliche USB-Buchsen mit hoher Lebensdauer für stark beanspruchende Anwendungen in rauen Einsatzumgebungen zu finden. Weitere Informationen über die Produktserie RUGGEDrive™ und mobile Speicherlösungen von Datakey finden Sie unter www.datakey.com.

ZUSAMMENFASSUNG

Dieses Whitepaper liefert einen kurzen Abriss vieler Themen, die bei der Implementierung von eingebetteten Systemen mit USB-Wechseldatenträger-Unterstützung zu beachten sind. Nach einem Blick auf die allgemeine Systemkonzeptionierung folgen viele praktische Beispiele für Implementierungsoptionen, die sich auf Hardware-, Systemsoftware- und Anwendungsebene bieten. Wir hoffen, dass dieses Whitepaper Ihre Entwicklungsarbeit erleichtert und Ihnen hilft, viele der typischen Schwierigkeiten zu vermeiden.

GLOSSAR

EHCI: Abkürzung für Enhanced Host Controller Interface. Hinter EHCI verbirgt sich eine Schnittstelle auf Registerebene für USB 2.0 Host-Controller. Die EHCI-Spezifikation ist zu finden unter: www.intel.com/content/www/us/en/io/universal-serial-bus/ehci-specification.html

OHCI: Abkürzung für Open Host Controller Interface. Hinter OHCI verbirgt sich eine Schnittstelle auf Registerebene für USB 1.1 Host-Controller. Die OHCI-Spezifikation ist zu finden unter: ftp://ftp.compaq.com/pub/supportinformation/papers/hcir1_0a.pdf

Gerätetreiber: Eine Softwarekomponente, die eine Reihe vordefinierter Funktionen implementiert und es dem Betriebssystem ermöglicht, in einheitlicher Weise mit Hardwaregeräten zu kommunizieren.

Klassentreiber: Ein Gerätetreiber für USB-Geräte, der eine geräteunabhängige Schnittstelle zu einem Subsystem des Betriebssystems bereitstellt.

Protokollstapel: Eine Sammlung an Softwarekomponenten, die ein Kommunikationsprotokoll implementiert.

Middleware: Eine Sammlung an Soft- bzw. Firmware, die eine Softwarefunktion bereitstellt, welche das Betriebssystem nicht bereitstellen kann.

RTOS: Abkürzung für Real Time Operating System. Ein RTOS bzw. Echtzeitbetriebssystem ist ein spezielles Betriebssystem eigens für Systeme, die in Bezug auf Determinismus, Durchsatz und/oder Antwortzeit spezielle Anforderungen stellen.

QUELLEN

[1] USB – Übersicht Universal Serial Bus, Intel-Website, http://www.intel.com/content/www/us/en/io/universal-serial-bus/universal-serial-bus.html
[2] USB-IF Soft- und Hardware-Tools, Website des USB-IF, www.usb.org/developers/tools
[3] USB-IF Compliance-Programm, Website des USB-IF, www.usb.org/developers/compliance
[4] NXP Schnittstellen- und Konnektivitätsprodukte, NXP-Website, www.nxp.com/products/interface_and_connectivity/usb_host_controllers/#products
[5] Cypress USB-Hosts, Cypress-Website, http://www.cypress.com/?id=186
[6] FTDI Produkte, FTDI-Website, www.ftdichip.com/Products/ICs.htm
[7] Tiva C-Serie MCUs, TI-Website, www.ti.com/lsds/ti/microcontroller/tiva_arm_cortex/c_series/overview.page
[8] TI Safety Mikrocontroller, TI-Website, www.ti.com/lsds/ti/microcontroller/safety_mcu/overview.page
[9] TI Sitara Mikroprozessoren, TI-Website, www.ti.com/lsds/ti/arm/sitara_arm_cortex_a_processor/overview.page
[10] TI OMAP Anwendungsprozessoren, TI-Website, www.ti.com/lsds/ti/omap-applications-processors/overview.page
[11] Freescale Kinetis Mikrocontroller, Freescale-Website, www.freescale.com/webapp/sps/site/homepage.isp?code=KINETIS
[12] Freescale Vybrid Mikroprozessoren, Freescale-Website, www.freescale.com/webapp/sps/site/homepage.isp?code=VYBRID
[13] Freescale i.MX-Serie Mikroprozessoren, Freescale-Website, www.freescale.com/webapp/sps/site/homepage.jsp?code=IMX HOME
[14] Atmel AVR Mikrocontroller, Atmel-Website, www.atmel.com/products/microcontrollers/avr/default.aspx
[15] Atmel ARM Mikrocontroller, Atmel-Website, www.atmel.com/products/microcontrollers/arm/default.aspx
[16] NXP LPC Mikrocontroller, NXP-Website, www.nxp.com/products/microcontrollers
[17] Microchip PIC Mikrocontroller, Microchip-Website, http://www.microchip.com/pagehandler/en-us/technology/usb/microcontrollers/home.html
[18] Windows Embedded, Microsoft-Website, www.microsoft.com/windowsembedded
[19] emUSB-Host, Segger-Website, www.segger.com/emusb-host.html
[20] Keil MDK-ARM, Keil-Website, www.keil.com/arm/mdk.asp
[21] Green Hills Produkte, Green Hills-Website, www.ghs.com/products.html
[22] MQX Tools, Freescale-Website, www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MQX
[23] TivaWare Ressourcen, TI-Website, www.ti.com/lsds/ti/microcontroller/tiva_arm_cortex/c_series/tools_software.page#tivaware
[24] ASM USB-Host-Stack, Atmel-Website, www.atmel.com/lmages/doc8486.pdf
[25] MQX Softwarelösungen, Freescale-Website, www.freescale.com/webapp/sps/site/homepage.jsp?code=MQX HOME
[26] USB-Spezifikation V2.0, Website des USB-IF, www.usb.org/developers/docs/usb20_docs
[27] On-The-Go und Embedded Host, Website des USB-IF, www.usb.org/developers/onthego
[28] SVS Reset-Verzögerung, TI-Website, www.ti.com/lsds/ti/power-management/supervisor-reset-ic-overview.page
[29] Hi-Speed USB PCB-Layout, Cypress-Website, http://www.cypress.com/?doclD=47409
[30] Saturn PCB-Toolkit, Saturn-Website, www.saturnpcb.com/pcb toolkit.htm
[31] Leiterbahn-Rechner, Advanced PCB-Website, http://www.4pcb.com/trace-width-calculator.html
[32] mBED-HDK, mbed-Website, http://mbed.org/teams/mbed/code/mbed-HDK/
[33] MSCUsbHost-Beispiel, mbed-Website, www.mbed.org/users/igorsk/code/MSCUsbHost
[34] FAT-Dateisystem Lizenzprogramm, Microsoft-Website, http://www.microsoft.com/en-us/legal/lntellectualProperty/IPLicensing/Programs/Default.aspx

Telefon: +49 211 959 87974
E-Mail: info@nexusindustrialmemory.com
Web: www.nexusindustrialmemory.com/de

NEXUS INDUSTRIAL MEMORY