



Digitale Hardwarechnittstellen

Teil 5: USB





USB (Universal Serial Bus) ist ein seriellcs Bussystem zur Verbindung externer Geräte (Maus, Tastatur, Drucker, Scanner, Speichermedien, Digitalkameras, Etikettendrucker, Barcodescanner, Teleskope, Maschinen u.v.m.) mit einem PC. Die Bits werden bei USB seriell (also nacheinander) und asynchron (also ohne separates Taktsignal) übertragen.

Geschichte, Standards, Geschwindigkeiten

Sehr viele Jahre gab es bei PCs (angefangen beim ersten PC, dem Apple I im Jahr 1976, dem Commodore PET und dem Tandy TRS-80 Model 1 im Jahr 1977 sowie dem IBM-PC und den sogenannten IBM-PC-Kompatiblen ab 1981) für jedes anzuschließende Gerät jeweils eine eigene Schnittstelle und Buchsenform. Drucker wurden über eine serielle oder parallele Schnittstelle angeschlossen, die Tastatur wurde ebenso wie die sich ab 1983 verbreitende Computermouse mit einem speziellen Stecker in die entsprechende Buchse am PC angeschlossen. Messgeräte, Modems, Dongles, Scanner etc. wurden lange Zeit über eine serielle Schnittstelle angeschlossen und für den Anschluss eines Bildschirms hatte jeder PC eine spezialisierte Buchse.

Für jedes Gerät und teilweise sogar für jedes Programm musste eine entsprechende Verbindungssoftware, ein sogenannter Treiber, vorhanden sein und die Geräte mussten meist schon beim Starten des PCs eingesteckt sein, um richtig zu funktionieren.

Um der wachsenden Vielfalt an Schnittstellen mit unterschiedlichen Stecker-/Buchsensystemen Herr zu werden und den Umgang mit PCs und Peripherie zu erleichtern und zu verbessern, wurde USB von einem Firmenkonsortium (IBM, Microsoft, Intel, Hewlett-Packard, Compaq, NEC u. a.) entwickelt und seit 1996 in PCs eingesetzt. Seit diesem Zeitpunkt können Tastaturen, Mäuse, Drucker, Scanner usw. über den universellen seriellen Bus (USB) angeschlossen werden, Geräte können auch im laufenden Betrieb verbunden werden (Hot Plugging) und werden vom Betriebssystem automatisch erkannt. Im Laufe der Jahre wurde USB ständig weiterentwickelt (Tabelle 1) und dabei wurden auch höhere Übertragungsgeschwindigkeiten eingeführt, die auch die Übertragung größerer Datenmengen (zum Beispiel bei externen Speichermedien) ermöglichen.

USB-Standards

USB-Standard	Jahr	Maximale Übertragungsrate
USB 1.0	1/1996	Low Speed (1,5 Mbit/s)
USB 1.1	8/1998	Full Speed (12 Mbit/s)
USB 2.0	4/2000	High Speed (480 Mbit/s)
USB 3.0 bzw. USB 3.1 Gen 1	11/2008	Super Speed (5 Gbit/s)
USB 3.1 bzw. USB 3.1 Gen 2	7/2013	Super Speed+ (10 Gbit/s)

Tabelle 1

Die Nennspannung von USB ist 5 Volt. Die Spannungsversorgung von Geräten mit geringer Stromaufnahme über die USB-Schnittstelle ist seit USB 1.0 möglich. Um auch höhere Stromstärken und inzwischen sogar höhere Spannungen zu ermöglichen, wurden die USB-Spezifikationen nach und nach erweitert ([1], Tabelle 2). Die mögliche Stromstärke und Spannung muss ein Gerät zunächst mit dem Host-Controller (PC) „aushandeln“.

USB-Buchsen am PC werden als solche meistens mit einem dem USB-Standard entsprechenden Symbol gekennzeichnet (Bild 1).

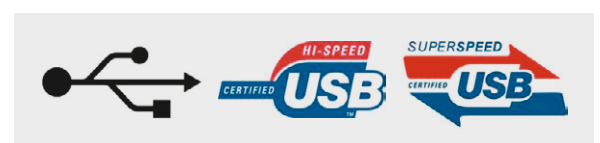


Bild 1: USB-Symbol (links), USB Hi-Speed (Mitte), USB Super Speed (rechts)

Spannungen und Stromstärken der USB-Standards

USB-Standard	Spannung	Max. Stromstärke	Max. Leistung
USB 1.0/1.1 (Low-Powered Port)	5 V	0,1 A	0,5 W
USB 2.0 (High-Powered Port)	5 V	0,5 A	2,5 W
USB 3.0/3.1	5 V	0,9 A	4,5 W
USB-BC 1.2 (Battery Charging)	5 V	1,5 A	7,5 W
USB-Typ C	5 V	3 A	15 W
USB-PD (Power Delivery)	5, 12 oder 20 V	5 A	100 W

Tabelle 2

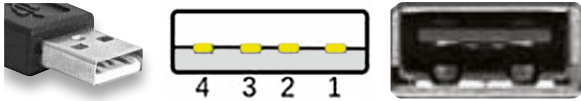


Bild 2: USB-Stecker Typ A, Belegung, USB-Buchse Typ A
Quelle Belegungen: <https://en.wikipedia.org/wiki/USB>

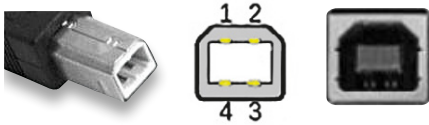


Bild 3: USB-Stecker Typ B, Belegung, USB-Buchse Typ B



Bild 4: USB-Stecker Typ Mini-B, Belegung, USB-Buchse Typ Mini-B

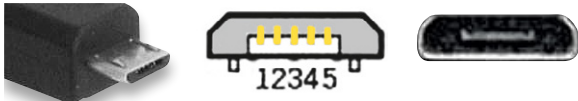


Bild 5: USB-Stecker Typ Micro-B, Belegung, USB-Buchse Typ Micro-B

Pin	Name	Aderfarbe	Funktion
1	V _{BUS}	Rot	+5 V
2	D-	Weiß	Data-
3	D+	Grün	Data+
4	GND	Schwarz	Ground

Bild 6: Pinout Typ A, Typ B

Pin	Name	Aderfarbe	Funktion
1	V _{BUS}	Rot	+5 V
2	D-	Weiß	Data-
3	D+	Grün	Data+
4	ID	No wire	On-The-Go ID • An Gnd: "A"-Stecker (Host) • Unverbunden: "B"-Stecker (Gerät)
5	GND	Schwarz	Ground

Bild 7: Pinout Mini/Micro Typ A, Typ B

Stecker, Buchsen, Kabel, Farben

Ein wesentlicher Vorteil von USB ist die Standardisierung der Stecker, Buchsen und Kabel. Der Standardsteckertyp heißt USB-Stecker Typ A. Bild 2 zeigt die Belegung der Pins von vorne betrachtet, USB-Logo oben. Die Belegung der Pins ist: 1: Vcc, 2: D-, 3: D+, 4: Gnd. Die Anschlüsse für die Betriebsspannung (Vcc und Gnd) sind leicht vorseilend (etwas länger) und werden daher beim Einstecken elektrisch verbunden, bevor die Datenleitungen (D+ und D-) verbunden werden. Per Konvention sind beim Kabel die Aderfarben: Rot für Vcc (+5 V), Weiß für Data- (D-), Grün für Data+ (D+) und Schwarz für Masse (Gnd) (Bild 6). USB-Stecker Typ A werden gemäß USB-Standard an den USB-Host (PC) angesteckt. Jeder moderne PC hat mehrere Typ A-Buchsen.

Als Gegenstück – zum Einstecken in den Drucker oder Scanner – gibt es den USB-Stecker Typ-B (Bild 3). Die Pinbelegung entspricht in der Nummerierung dem Typ-A-Stecker.

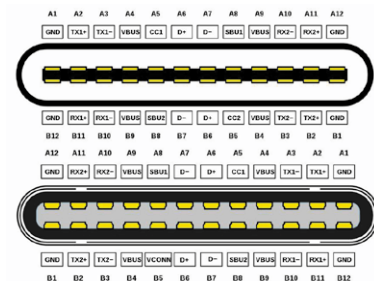
Viele Geräte (zum Beispiel Mobiltelefone, Digitalkameras oder Card-reader) verwenden USB-Stecker Typ Mini-B (Bild 4) oder Micro-B (Bild 5), bei denen zusätzlich zu Vcc (1, rote Aderfarbe), Gnd (5, schwarze Ader), D- (2, weiß) und D+ (3, grün) noch ein Pin 4 (ID, On-the-Go) zur Differenzierung zwischen Typ A (Host, Pin an Gnd) und Typ B (Device, Pin nicht verbunden) ermöglicht wird (Bild 7).

Im August 2014 wurde die Spezifikation für einen neuen Steckertyp USB Typ C (Bild 8) verabschiedet, der punktsymmetrisch ist und in beiden Orientierungen eingesteckt werden kann. (Es gibt zwar auch Typ-A-Stecker, die sich in beiden Orientierungen einstecken lassen, diese entsprechen aber keinesfalls dem Standard.) Bild 8 zeigt, wie es realisiert ist, dass die Typ-C-Stecker in beide Orientierungen gesteckt werden können: Die Pins sind symmetrisch auf beiden Seiten herausgeführt. USB-Stecker Typ C unterstützen alle bisherigen Übertragungsspezifikationen bis einschließlich USB 3.1 mit bis zu 10 Gbit/s und USB Power Delivery mit maximal 100 W. Umgekehrt unterstützt ein Gerät, das Stecker Typ C aufnimmt, nicht notwendigermaßen auch USB 3.1 oder USB Power Delivery. Ein Typ-C-Stecker kann an einen USB-Host und an USB-Geräte angeschlossen werden. Mit USB 3.0 wurden auch neue USB-Typ-B-Stecker/-Buchsen und USB-Typ-Micro-B-Stecker/-Buchsen definiert. Langfristig soll USB Typ C die verschiedenen USB-Steckertypen ablösen.

Über ein USB-Typ-C-Kabel ist die Übertragung von Daten, Audio, Video und Ladestrom möglich. An beiden Seiten befinden sich die gleichen verdrehsicheren Typ-C-Stecker. Man benötigt dann nicht mehr die Unterscheidung von Steckern am Host (Typ A) und Steckern am Gerät (Typ B), und mit Typ C wird USB 3.1 Gen 2 mit voller Geschwindigkeit unterstützt. Bevor es so weit ist, verwenden manche Hersteller verschiedene Farben an Buchsen, um auf die Möglichkeiten eines USB-Anschlusses hinzuweisen (Tabelle 3).



Bild 8: USB-Stecker Typ C, Belegung Typ C (Stecker, Buchse)
Quelle Belegungen: <https://en.wikipedia.org/wiki/USB-C>



Farben von USB-Typ A-Anschlüssen (nicht im USB-Standard vorgeschrieben)

Farbe des USB-Anschlusses	Bedeutung
Schwarz oder Weiß (oder nur USB-Symbol)	USB 2.0, für alle USB-Geräte (Maus, Tastatur, USB-Stick etc.), aber nur bis High Speed
Blau (oder SS für Super Speed)	USB 3.0, volle Unterstützung für schnelle USB 3.0 Geräte wie externe Festplatten oder USB-Sticks
Gelb (oder Blitz)	USB-Anschluss mit Dauerspannung, auch wenn der PC/Laptop nicht eingeschaltet (aber an Spannung) ist
Rot oder Orange	USB 2.0 oder USB 3.0 oder hohe Stromstärke je nach PC-Hersteller

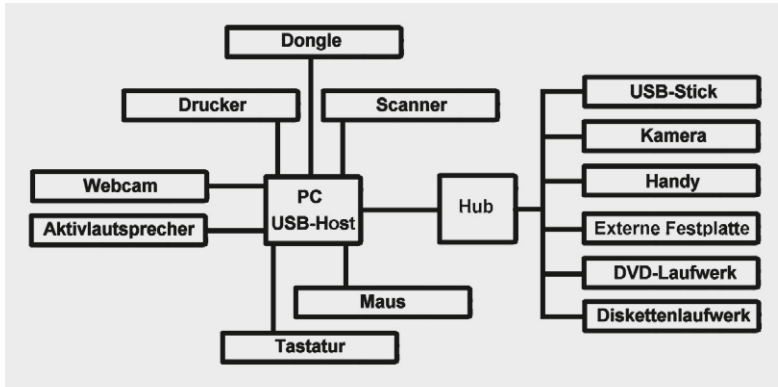


Bild 9: USB-Topologie

Busaufbau

Im Mittelpunkt eines USB-Systems, das sternförmig aufgebaut ist, steht ein USB-Host (PC, Root Hub), der bis zu 127 angeschlossene Geräte (Nodes, Slaves, Endgeräte) steuern kann. Bild 9 zeigt einige typische, an einen PC angeschlossene Geräte: Tastatur, Maus und Drucker sind selbstverständlich. Aber auch eine Webcam, ein Aktivlautsprecher oder ein Dongle können über USB angeschlossen sein. Statt eines Nodes kann auch ein sogenannter Hub am USB-Host angeschlossen sein, über den wiederum weitere Nodes wie USB-Sticks, externe Festplatten usw. angeschlossen sein können. An einen Hub könnte auch wiederum ein Hub angeschlossen werden. Die maximale Leitungslänge beträgt ca. fünf Meter. Die Kommunikation erfolgt immer vom USB-Host zu den Nodes. Eine Ausnahme von dieser Regel wurde erst durch eine Ergänzung (mit USB 2.0) mit USB-on-the-Go (OTG) eingeführt (Elektronikwissen).

Man kann jedes Gerät an eine beliebige der vorhandenen USB-Buchsen stecken, und es gibt bei USB Standard-Treiber, durch die man zum Beispiel Tastaturen, Mäuse, Massenspeicher, Digitalkameras, Scanner usw. an einen PC anschließen kann, ohne spezialisierte Treiber für das Betriebssystem installieren zu müssen.

Neben der Spannungsversorgung (+5 Volt und Gnd) gibt es wie oben erwähnt die zwei Datenleitungen D+ und D-. Ein angeschlossener Node kann dem Host die gewünschte Übertragungsgeschwindigkeit signalisieren, indem die Leitung D+ (für Full Speed) bzw. die Leitung D- (für Low Speed) mit einem Pull-up-Widerstand nach 3,3 V gezogen wird (Bild 10). Der Host baut die Verbindung mit der entsprechenden Übertragungsrate auf. Ein High-Speed-Gerät verhält sich zunächst wie ein Full-Speed-Gerät (D+ über Widerstand an 3,3 V) und identifiziert sich danach durch Abschalten des Pull-up-Widerstands und Aktivierung von 45 Ohm Abschlusswiderständen an D+ und D-.

Auf den Datenleitungen D+ bzw. D- gibt es die Spannungspegel 0 V und 3,3 V. Die elektrische Übertragung erfolgt bei USB auf den Leitungen D+ und D- symmetrisch: Eine Leitung führt das jeweils inverse Signal zur anderen. Außer D+ und D- gibt es noch die Versorgungsspannung +5 V und Gnd.

Adressierung

Der Host-Controller adressiert die angeschlossenen Geräte mittels einer 7 Bit langen Kennung. Daher ist die Adressierung von $2^7 = 127$ Geräten an einem Host möglich. Wenn ein Gerät an einen PC angeschlossen

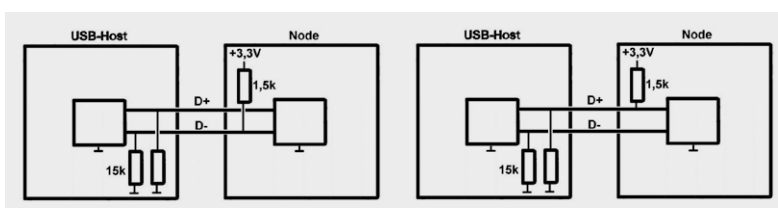


Bild 10: Bei Low-Speed-Geräten (links) ist der Pull-up-Widerstand an D-, bei Full-Speed-Geräten (rechts) ist der Pull-up-Widerstand an D+.

wird, dann wird das vom PC (USB-Host) erkannt, der Host schaltet den entsprechenden Port ein und legt beide Datenleitungen für mindestens 10 ms auf Gnd, wodurch der Slave die Adresse 0 belegt und dann vom Host eine neue Adresse zugewiesen bekommt. Danach liest der Host Hersteller- und Produkt-Id des Slaves und kann dadurch den richtigen Treiber aktivieren.

Datenübertragung

Wenn kein Gerät eingesteckt ist, dann „sieht“ der Host beide Datenadern (D+ und D-) als Low, weil der Host die beiden Adern mit 15 kΩ nach Gnd (Bild 10) zieht („Detached“). Wenn ein Gerät eingesteckt wird, dann „sieht“ der Host entweder D+ (bei einem Full- oder High-Speed-Gerät) oder D- (bei einem Low-Speed-Gerät) als High („Attached“) (Bild 10). Der Zustand, dass eine Datenader nach High gezogen wird und die andere Datenader auf Low liegt, nennt sich „Idle“ (Ruhezustand). Vor und nach jeder Paketübertragung liegt der Idle-Zustand vor.

In vielen Publikationen wird bezüglich der Spannungslevel an den Datenleitungen vom J-Zustand und vom K-Zustand gesprochen. Der J-Zustand ist gleich mit dem Idle-Zustand, also wenn eine der Datenadern auf High gezogen ist und die andere Datenader Low ist. Der K-Zustand ist das Gegenteil

USB OTG

Ein Android-Gerät (Tablet oder Smartphone) wird normalerweise per USB-Kabel an einen PC angeschlossen, der dann das Hostsystem darstellt, während das Android-Gerät die Funktion eines Clients wahrnimmt. So verbunden lassen sich beispielsweise Fotos oder andere Dateien zwischen PC und Android-Gerät übertragen. Möchte man – zum Beispiel unterwegs im Urlaub – Fotos vom Smartphone oder Tablet auf einen USB-Stick übertragen, ohne den Umweg über einen PC zu gehen, dann muss man sich des USB-OTG-Standards bedienen. OTG steht für „On-the-Go“ und ermöglicht die direkte Verbindung von Smartphone oder Tablet zu einem USB-Stick, einer Digitalkamera oder Maus/Tastatur. Das Android-Gerät agiert nun als Host. Zwei Dinge müssen für USB OTG erfüllt sein:

1. Das Android-Gerät (und das angeschlossene Gerät!) muss den USB-OTG-Standard unterstützen, was man beispielsweise unter <http://www.gsmarena.com/results.php3> (Stichwort: USB-Host) überprüfen kann.
2. Die Verbindung zwischen Android-Gerät und Gerät muss mit einem USB-OTG-Kabel hergestellt werden, das sich günstig erwerben lässt.

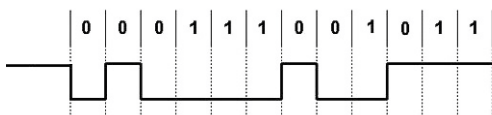


**USB Spannungspegel/Bus-Zustände**

Bus-Zustand	Spannungspegel
Differenzielle „1“	D+ High, D- Low
Differenzielle „0“	D- High, D+ Low
Single Ended Zero (SE0)	D+ und D- Low
Single Ended One (SE1)	D+ und D- High
Daten J-Zustand: Low Speed Full Speed	Differenzielle „0“ Differenzielle „1“
Daten K-Zustand: Low Speed Full Speed	Differenzielle „1“ Differenzielle „0“
Idle Zustand: Low Speed Full Speed	D- High und D+ Low D+ High und D- Low
Resume-Zustand	K-Zustand
Start of Package (SOP)	Datenadern gehen von Idle auf K-Zustand
End Of Package (EOP)	SE0 für 2 Bit Zeit, dann J-Zustand für 1 Bit Zeit
Disconnect	SE0 für $\geq 2\mu\text{s}$
Connect	Idle für $2,5\mu\text{s}$
Reset	SE0 für $\geq 2,5\mu\text{s}$

vom J-Zustand. Von J und K spricht man, weil die Spannungsverhältnisse bei Full Speed und Low Speed gegenteilig sind. Wenn beide Datenadern auf Low gezogen sind, wird von „Single Ended Zero“ (SE0) gesprochen. Beide Datenadern auf High nennt sich „Single Ended One“ (SE1) und darf nie vorkommen. Wenn ein Host eine Kommunikation mit einem Gerät starten möchte, dann setzt er eine Reset-Bedingung, die darin besteht, dass der Host beide Datenadern für mindestens 10 ms auf Low zieht (SE0). Ein Gerät kann die Reset-Bedingung nach $2,5\mu\text{s}$ erkennen. [Tabelle 4](#) fasst einige Bus-Zustände zusammen.

USB ist eine serielle Datenübertragung. Die Daten werden durch eine differenzielle Darstellung von Bits übertragen, das heißt, wenn D+ High (3,3 V) ist, dann ist D- Low (0 V) und umgekehrt ([Bild 11](#)). Da der Empfänger die Differenzspannung beider Datenleitungen bildet, ist die Übertragung sehr störsicher.



USB benutzt zur Datenübertragung NRZI (=Non-Return-to-Zero-Inverted). Dabei wird eine 0 durch einen Wechsel des Signal-Levels dargestellt, während zur Darstellung einer 1 kein Wechsel des Signal-Levels erfolgt. Zur Verdeutlichung: Bei einer Folge mehrerer

Nullen hintereinander erfolgt jedes Mal ein Levelwechsel High $\leftrightarrow$ Low. Eine Folge mehrerer Einsen hintereinander wird durch das Halten des Signal-Levels dargestellt ([Vgl. auch Bilder 11 und 14](#)).

Damit auch bei einer Abfolge vieler aufeinanderfolgender Einsen ein korrekter Signalwechsel sichergestellt ist, wird bei der Übertragung eines USB-Paketes vor der NRZI-Kodierung das sogenannte Bit-Stuffing eingesetzt, was bedeutet, dass nach sechs aufeinanderfolgenden Einsen eine Null in den Datenstrom eingefügt wird, bevor das Signal NRZI-verschlüsselt wird. Dadurch ergibt sich mindestens alle sieben Bit-Zeiten ein Signalwechsel und die Synchronisation zwischen Sender und Empfänger bleibt erhalten. Bit-Stuffing wird nach dem Sync-Muster aktiviert ([Vgl. auch \[2\]](#)). Im Sender werden die seriellen Daten zunächst dem Bit-Stuffing und dann dem NRZI-Encoding unterzogen und schließlich in das differenzielle Spannungssignal umgesetzt und auf die Datenleitungen D+ und D- gegeben. Im Empfänger ist die Reihenfolge umgekehrt: Das differenzielle Signal wird gelesen und dann zunächst durch den NRZI-Dekoder und dann den Bit-Destuffer verarbeitet. Der Takt für die Abtastung wird aus einem Synchronisier-Bitmuster und aus dem Datenstrom selbst generiert.

Die Kommunikation bei USB wird von einem Host-Controller (meistens auf dem Motherboard eines PCs) gesteuert und erfolgt in Datenpaketen. Jedes übertragene Datenpaket beginnt mit einem Synchronisationsfeld (SYNC, in [Bild 11](#) als 00000001 zu sehen), das zur Synchronisation dient. (Es gibt ja keine separate Taktleitung.) Das SYNC-Feld ist bei Low Speed und Full Speed 8 Bit lang und bei High Speed ist es 32 Bit lang (31 mal „0“ und eine „1“). Die letzten zwei Bit zeigen an, wo das nachfolgende PID-Feld beginnt. Teil dieses SYNC-Feldes ist der „Start Of Packet“ (=SOP)-Begrenzer. Am Schluss eines Datenpaketes wird ein „End of Package“ (=EOP)-Begrenzer übertragen. Nach dem SYNC wird der Packet Identifier (= PID) übertragen, der aus zwei 4-Bit-Feldern besteht (z. B.: 1010 0101, [vgl. Bilder 11 und 12](#)). Die ersten vier Bits des PID stellen den Pakettyp dar und die zweiten vier Bit stellen das Einerkomplement der ersten Bits dar und ermöglichen somit eine Überprüfung der Fehlerfreiheit der ersten vier Bits ([Bild 13](#)). Die im Beispiel übertragene PID &b0101 steht für „Start of Frame“ ([Tabelle 5](#)) und wird vom Protokolldekorator des Logikanalysators entsprechend als „PID SOF“ dekodiert ([Bild 12](#)). Beim PID SOF wird als Nächstes die Framenummer und dann eine CRC5-Prüfnummer übertragen.

Im Beispiel ([Bilder 11 und 12](#)) wird insgesamt also ein „Start of Frame“-Packet übertragen, das aus SYNC, PID, Framenummer, CRC5 und EOP besteht. Mit jedem Frame wird eine 11-Bit-Framenummer übertragen, die pro Frame vom Host um 1 erhöht wird ([Bild 12](#)). Es gibt bei USB vier verschiedene Paket-Typen ([Tabelle 5](#)). Eine Darstellung für PID IN zeigt [Bild 14](#). Auch hier sieht man wieder den Aufbau mit SYNC, PID, Adresse/Endpunkt, CRC5 und EOP.

Kommunikation mittels UART-USB-Konverter

Die USB-Schnittstelle ist für den Benutzer eines PCs sehr einfach zu benutzen: Anschließendes Gerät auswählen, mit passendem USB-Kabel mit dem PC verbinden, Treiber vom Betriebssystem installieren bzw. aktivieren lassen und die Kommunikation kann beginnen. Aufgrund der Komplexität des USB-Protokolls ist eine Verwendung in selbst programmierten Mikrocontroller-Anwendungen allerdings recht schwierig und wird auch durch die wenigsten Mikrocontroller unterstützt. Um dennoch eine Verbindung zwischen einem PC mit einem USB-Anschluss und



Bild 11: Logikanalysatordarstellung (PID SOF)



Spannungen und Stromstärken der USB-Standards

PID-Gruppe	PID-Wert	PID-Name	Beschreibung
Token	0001	OUT	Host möchte senden. Adresse + Endpunktnummer im Host -> Endgerät-Transition
	1001	IN	Host möchte lesen. Adresse + Endpunktnummer im Endgerät -> Host-Transition
	0101	SOF	„Start Of Frame“-Markierung und Frame-Nummer
	1101	SETUP	Adresse + Endpunktnummer im Host -> Endgerät-Setup-Transaktion zu einem Kontroll-Endpunkt
	Format: Sync PID ADDR ENDP CRC5 EOP bzw. Sync SOF Framenummer CRC5 EOP		
Data	0011	DATA0	Datenpaket, gerade Parität
	0111	DATA1	Datenpaket, ungerade Parität
Format: Sync PID Data (0-1023 Byte) CRC16 EOP			
Handshake	0010	ACK	Empfänger bestätigt fehlerfreien Empfang
	1010	NAK	Empfänger akzeptiert Daten nicht oder Sender kann nicht übermitteln
	1110	STALL	Endpunkt ist „abgestürzt“
	Format: Sync PID EOP		
Special	1100	PRE	Host-initiierte Ankündigung: schaltet Downstream-Bus-Übertragung für langsame USB-Geräte ein.

Tabelle 5

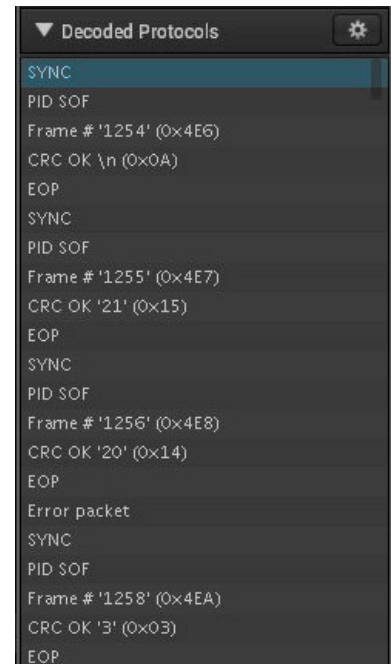


Bild 12: Dekodiertes USB-Signal

einer – evtl. selbst erstellten – Mikrocontroller-Schaltung möglich zu machen, gibt es zum Glück sehr kleine und sehr einfach zu benutzende Konvertier-ICs, die mit wenigen Bauteilen Zusatzbeschaltung eine USB-Schnittstelle auf eine bidirektionale serielle UART-Schnittstelle umsetzen (Bild 15).

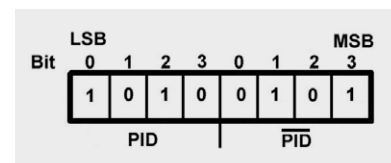


Bild 13: PID = Packet Identifier



Bild 14: Logikanalysatordarstellung (PID IN)

Beispiele für USB-UART-Konverter-ICs:

- CP2102, CP2102N (neuer) von Silicon Labs (vgl. [3])
- FT232R von Future Technology International Ltd (FTDI) (vgl. [4])
- PL-2303HXD bzw. PL-2303RA von Prolific (vgl. [5])

ELV bietet einige dieser ICs mit benötigter Zusatzbeschaltung (Bild 16) auf sehr kleinen Platinen an, die zwischen eine Mikrocontrollerschaltung und einen PC geschaltet werden. Der PC „sieht“ quasi eine USB-Verbindung und der Mikrocontroller „sieht“ eine UART-Schnittstelle, die von jedem Mikrocontroller unterstützt wird und sehr einfach benutzt werden kann.

- USB-Modul UM2102N, Komplettbausatz (CO-15 09 52)
- Mini-USB-Modul UM2102, Komplettbausatz (CO-09 18 59)

Der Betrieb dieser Module ist an 5 V und an 3,3 V möglich. Die Module beziehen ihre Spannungsversorgung über die USB-Buchse. Es gibt sogar zwei Spannungsausgänge für die Spannungsversorgung eigener Schaltungen (5 V/500 mA bzw. 3,3 V/100 mA). Es werden Übertragungsraten von 300 bis 3.000.000 Baud unterstützt. Durch die kleinen Ausmaße (28 x 18 mm) und das 2,54-mm-Raster lassen sich die Module in eigene Schaltungen integrieren oder mit einem Mini-Kunststoffgehäuse (DR03, hellgrau, CO-07 60 12) als eigenständiges Modul benutzen.

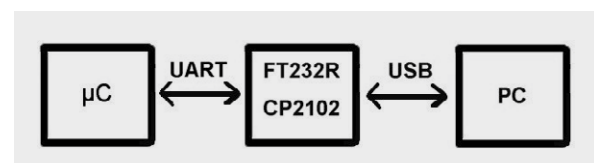


Bild 15: USB-Konverter

Der Anschluss eines derartigen USB-Moduls ist sehr einfach: Auf Mikrocontroller-Seite werden die benötigten Datenleitungen (RxD oder/und TxD) und Gnd verbunden (Bild 17) und zum PC wird einfach eine USB-Verbindung hergestellt. Das USB-Modul wird vom PC-Betriebssystem erkannt und auf dem PC kann man eine virtuelle serielle Schnittstelle benutzen. Es sind einfache Verbindungen mit einem sogenannten Terminalprogramm (wie zum Beispiel Hterm [6]) oder komplexere Verbindungen mit selbst erstellten oder erworbenen PC-Programmen möglich.

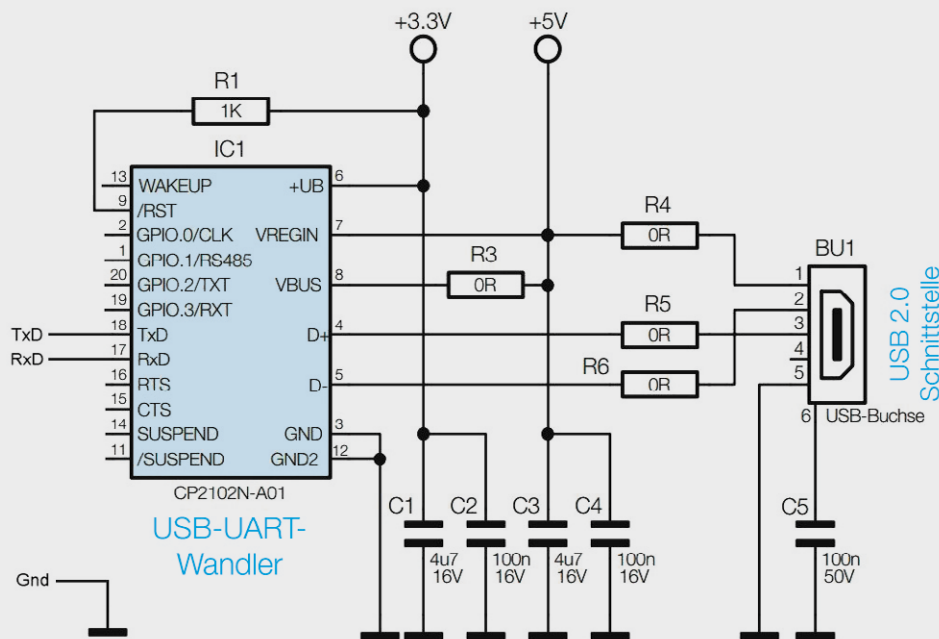


Bild 16:
Schaltplan (Aus-
schnitt) USB-Modul
UM2102N

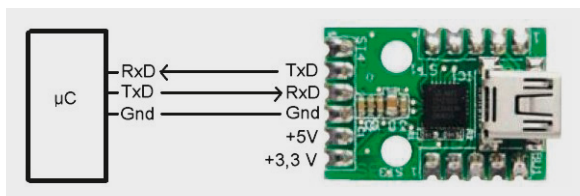


Bild 17: USB-UART-Umsetzer UM2102(N)

Fazit

Mit der Einführung von USB hatte die Vielfalt externer PC-Schnittstellen (seriell, parallel, Maus, Tastatur ...) ein Ende. Heutige PCs haben immer mehrere USB-Buchsen, in die diverse Geräte über standardisierte Kabel angeschlossen werden können. Ein ausgeklügeltes USB-Übertragungsprotokoll und eine hohe Übertragungsgeschwindigkeit bis zu 10 Gbit/s haben die USB-Schnittstelle zu einer wirklich universellen Verbindungsmöglichkeit gemacht.

Für die Anbindung eigener Mikrocontrolleranwendungen an PCs mit USB-Anschlüssen gibt es sehr preiswerte, sehr kleine USB-UART-Module, durch die der Programmierer sowohl auf der Mikrocontrollerseite als auch auf dem PC einfache serielle Befehle benutzen kann.



Weitere Infos:

- [1] USB-Informationen:
<https://en.wikipedia.org/wiki/USB>
https://de.wikipedia.org/wiki/Universal_Serial_Bus
<http://www.usb.org/developers/docs/ecn1.pdf>
- [2] https://www.tu-chemnitz.de/informatik/RA/news/stack/kompodium/vortraege_97/usb/datenfl.html
- [3] Datenblatt CP2102N von Silicon Labs:
<https://www.silabs.com/documents/public/data-sheets/cp2102n-datasheet.pdf>
Unterschiede CP2102 und CP2102N:
<https://www.silabs.com/documents/public/application-notes/an976-cp2102-3-4-9-to-cp2102n-porting-guide.pdf>
- [4] Datenblatt FT232R von FTDI:
http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf
- [5] Datenblatt PL-2303HXD von Prolific:
http://www.prolific.com.tw/UserFiles/files/ds_pl2303HXD_v1_4_4.pdf
Datenblatt PL-2303RA von Prolific:
http://prolificusa.com/files/DS_PL2303RA_d20120504.pdf
- [6] <http://www.der-hammer.info/terminal/>

Preisstellung Juni 2017 – aktuelle Preise im ELV Shop

Empfohlene Produkte	Best.-Nr.	Preis
USB-Modul UM2102N, Komplettbausatz	CO-15 09 52	€ 5,95
Mini-USB-Modul UM2102, Komplettbausatz	CO-09 18 59	€ 5,95
Mini-Kunststoffgehäuse DR03, hellgrau	CO-07 60 12	€ 3,95
ELV High Speed Mini USB Modul UM-FT2232H, Komplettbausatz	CO-09 93 47	€ 16,95
myAVR mySmartUSB MK2	CO-07 28 25	€ 27,95
Manhattan USB-Kabel, USB-Stecker (Typ A) auf Seriell-Stecker, P FTDI FT232RL-Chipsatz, 0,45 m	CO-12 99 00	€ 22,95
Manhattan USB-Kabel, USB-Stecker (Typ A) auf Seriell-Stecker, Prolific PL-2303HXD-Chipsatz, 0,45 m	CO-12 98 99	€ 11,95
Manhattan USB-Kabel, USB-Stecker (Typ A) auf Seriell-Stecker, Prolific PL-2303HXD-Chipsatz, 1,8 m	CO-12 99 61	€ 12,95
Manhattan USB-Kabel, USB-Stecker (Typ A) auf Seriell-Stecker, Prolific PL-2303RA-Chipsatz, 1,8 m	CO-12 99 62	€ 11,95