

Datenbusse

in elektronischen Schaltungen

Von Hannes Nordmann

Dieser Artikel beschäftigt sich mit einigen grundlegenden Unterschieden und den Vor- und Nachteilen verschiedener Bussysteme in elektronischen Schaltungen und vergleicht mehrere serielle Busse auf ihre Eignung in verschiedenen Systemen.

Wenn man das Wort Bus hört, denkt man unweigerlich an ein Verkehrsmittel, das Menschen von einem Punkt zum anderen befördert. Zwar geht es in diesem Artikel nicht um Autobusse, aber so ganz abwegig ist diese Überlegung nicht, denn sowohl mit einem Auto- als auch einem Datenbus wird etwas transportiert. Mit dem Autobus sind es Menschen, beim Datenbus sind es, wie es der Namen schon sagt, Daten beziehungsweise Informationen.

Ein Datenbus, der effektiv Daten an die Peripherie senden und auch empfangen kann, soll eine hohe Integrierbarkeit in andere Systeme (Geräte und deren Kombinationen) und (wenn gewünscht) eine hohe Anzahl von beteiligten Komponenten aufweisen. Kernfragen, die über die Eigenschaften des Bussystems entscheiden, wären:

- Wie schnell soll die Datenübertragung sein?
- Wie viele Komponenten sollen individuell ansprechbar sein?
- Soll die Hardware möglichst ökonomisch ausgelegt sein?

Datenbusse lassen sich in die Kategorien seriell/parallel und synchron (mit Takt)/asynchron (ohne Takt) einteilen.

Der parallele Bus

Der parallele Datenbus verfügt über wenigstens acht bidirektionale Datenleitungen. Heutzutage gibt es Mikrocontroller, die bis zu 32 Bit breite Datenbusse unterstützen und dadurch auch einen sehr hohen Datendurchsatz aufwei-

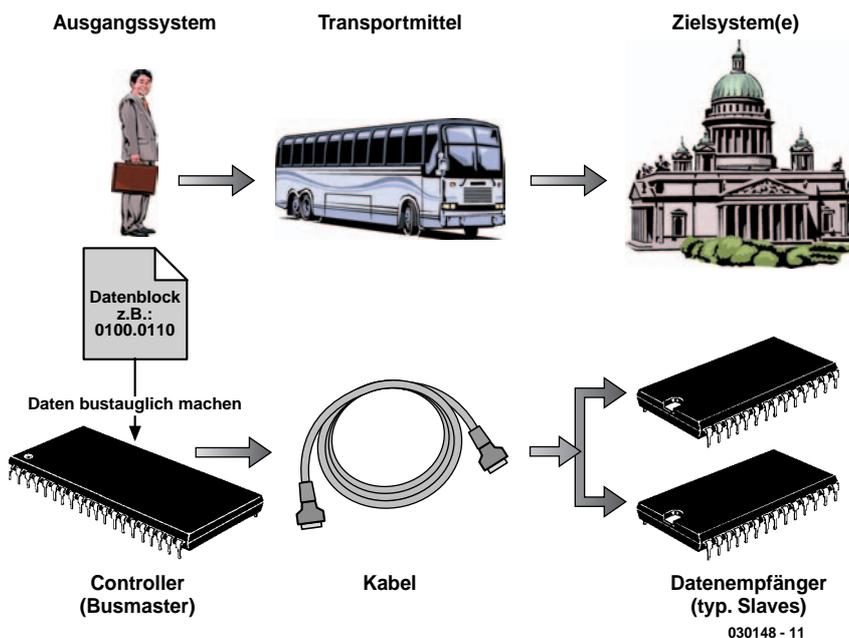


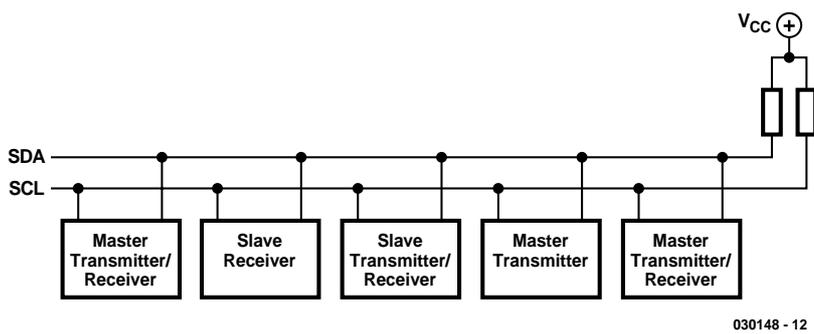
Bild 1. Jeder Bus ist ein Transportmittel.

sen. Eingesetzt werden solche Mikrocontroller in rechenintensiven Systemen, wie zum Beispiel im Security- und Entertainment-Bereich (DSPs).

Der serielle Bus

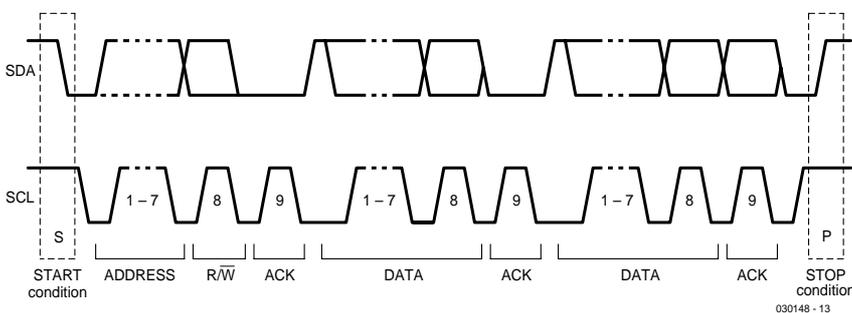
Beim seriellen Bus werden Daten seriell über meistens weniger als drei Leitungen gesendet und empfangen. Dies bedeutet, dass die Datenbits des zu übertragenden Wortes sukzessive auf Reisen gehen. In diesem Vergleich kann man

bereits erkennen, wo die Stärken und Schwächen beider Bussysteme liegen. Der serielle Bus weist - bedingt durch die zumeist einzige Datenleitung - eine langsame Datenübertragungsrate auf. Der parallele Bus ist in punkto Hardware sehr aufwendig, was wegen der vielen Anschlüsse besonders auf kleinen Platinenflächen Schwierigkeiten bereiten kann. Neben den reinen Datenleitungen gibt es in den meisten parallelen Bus-



030148 - 12

Bild 2. Physikalischer Aufbau eines I²C-Busses.



030148 - 13

Bild 3. Datenübertragung im I²C-Bus.

sen noch mehrere Steuerleitungen, zum Beispiel eine Taktleitung (manchmal auch als STROBE oder ENABLE bezeichnet), mit der alle Datenübertragungen synchronisiert werden. Bei asynchronen Bussen wird anstatt mit einem Taktsignal mit so genannten Handshakes gearbeitet.

Dabei legt der Sender die Daten auf den Bus und gibt mit einem Signal (Strobe) bekannt, dass der Empfänger die Datenleitungen abfragen kann. Der Empfänger übernimmt dann die Daten und sendet seinerseits ein Signal (Acknowledge) zur Bestätigung der Datenübernahme. Diese Art der asynchronen Datenübertragung nennt man auch *Verriegelte Kommunikation* (Interlocked Communication). Diese Form hat einen entscheidenden Vorteil gegenüber der taktsynchronen Übertragung: Sie ist durch den zahnradartigen „Dialog“ gegenüber Störungen hardware-technischer Art weitestgehend immun.

Microwire Bus

Trotz der verhältnismäßig geringen Leistungen sind taktsynchrone seri-

elle Busse die anspruchsvollsten. Es gibt sehr viele Arten taktsynchroner serieller Busse mit unterschiedlichen, genau auf den Verwendungszweck zugeschnittenen Spezifikationen.

Als Beispiel sei der Microwire-Bus genannt, der überwiegend zur seriellen Datenübertragung in kleineren EEPROMs benutzt wird (zum Beispiel der 93C46 – einen 64x16Bit Speicher). Finden kann man diesen Bus nur relativ selten in manchen Geräten der Unterhaltungselektronik, zum Beispiel als Programmspeicher in Satelliten-Receivern. Ein entscheidender Nachteil wäre die mangelnde Vielseitigkeit innerhalb eines Microwire-Busses, da die Auswahl des Empfänger-Chips (Slave) sehr hardware-lastig über je eine Mikrocontroller-Leitung erfolgt. Möchte man mit zehn Microwire-Chips kommunizieren, benötigt man auch zehn Chipselect-Leitungen. Neben der Chipselect- und der Taktleitung umfasst der Microwire-Bus zwei Datenverbindungen (IN und OUT) und gehört deswegen in die Familie der 4-Wire-Busse. Aus diesem Grund ist der Microwire-Bus fast aussch-

ließlich in Single-Master-Single-Slave-Systemen zu finden. Ein „klassisches“ Einsatzgebiet findet man bei EEPROMs in Telefonen, Fernsehapparaten oder anderen Konsumelektronikgeräten, die Werte speichern, die nicht oft verändert werden müssen, zum Beispiel Festwählnummern in Digitaltelefonen oder Daten zur HF-Abstimmung in Fernseh- oder Sat-Receiver Geräten.

I²C Bus

Ein seit gut zwanzig Jahren häufig in Unterhaltungselektronik anzutreffendes Bussystem stammt aus dem Hause Philips: der I²C-Bus (Inter Integrated Circuit). Dieser Bus besitzt zwei bidirektionale Daten- und Taktleitungen: SDA (Serial Data) und SCL (Serial Clock) mit denen er im 7-Bit Modus $2^7 = 128$ verschiedene Chips ansteuern kann (seit der Version 1.0 von 1992 gibt es auch 10-Bit-Adressen). In der generellen Struktur des I²C-Busses (Bild 2) sind alle Chips über SDA und SCL miteinander verbunden.

Der I²C-Bus kennt keine Slave-Auswahl über eine separate Leitung wie der Microwire-Bus, sondern eine protokollspezifische Adressstruktur, mit der jeder Chip im I²C-Bus allein und unverwechselbar angesteuert werden kann. Jeder Chip muss also eine unverwechselbare Adresse haben, über die nur er angesprochen werden kann.

Jeder serielle Bus wird mit einer Start- und Stoppsequenz initialisiert. Zwischen „Stopp“ und „Start“ ist der I²C-Bus inaktiv. Dabei führt die SDA-Leitung wegen der Pullup-Widerstände ein logisches High. Üblicherweise werden auch die Taktpulse von SCL gestoppt.

Die komplette Adressierung und Datenübertragung zwischen Master und Slave läuft folglich zwischen „Start“ und „Stopp“ ab. Damit während der Übertragung keine Fehler entstehen und sich in der Anwendung fortpflanzen können, hat man an das Ende eines jeden übertragenen Bytes ein Acknowledge (ACK) gehängt. Dieses ACK-Signal wird nach erfolgreicher Übertragung vom angesprochenen Slave, zum Master geschickt.

Bild 3 zeigt, dass hinter der 7-Bit Slave-Adresse ein weiteres Bit die Datenrichtung angibt. Denn die meisten I²C-Chips, wie der PCF8574 – ein Portexpander, können entweder eine lesende oder schreibende Rolle übernehmen. Im Beispiel heißt „Lesen“ nichts anderes als eine Datenabfrage an den Ports des Chips. Schreiben ist bei ihm gleichbedeutend mit dem Setzen von logischen Pegeln auf den Ports.

Der I²C-Bus ist hervorragend für den Datenaustausch zwischen den verschiedenen Mikro-Modulen in Unterhaltungs- und Kon-

sumelektroniken geeignet, da er mit seiner doch relativ großzügigen Anzahl von möglichen Busteilnehmern eine hohe Zahl von verschiedensten Sensoren und Aktoren erlaubt.

1-Wire Bus

Und dann gibt es noch einen 1-Wire-Bus (auch MicroLAN genannt) von Dallas Semiconductor, der nur eine einzige Datenleitung besitzt, über die alle Transaktionen durchgeführt werden. Selbst ihre Energie erhalten die 1-Wire-Chips über diese Datenleitung. So benötigt der 1-Wire Bus nur zwei Drähte (Datenleitung und natürlich Masse).

Diese Form der Energieübermittlung nennt man auch *parasitic power*. Der sehr simple Hardwareaufwand erfordert im Gegenzug eine sehr ausgeklügelte Übertragungsstruktur auf Softwareebene, um die sichere Übermittlung der Informationen gewährleisten zu können.

Der Trick: Jeder Chip besitzt eine eigene Zeitbasis, die die ankommenden Signale nach ihrem logischen Gehalt unterscheiden kann. Um eine logische 1 an den Chip zu senden, wird der Pegel der Datenleitung vom Master für maximal 15 μ s auf Masse gezogen. Eine logische 0 entspricht einem Low-Pegel von mindestens 60 μ s.

Der zeitliche Abstand zwischen beiden Zuständen ist mit Absicht so groß gewählt, um eventuelle zeitliche Verschiebungen (verursacht durch Produktstreuungen) weitestgehend überbrücken zu können (Worst-Case-Design). Der 1-Wire Bus ist also ein PWM-kodiertes Bussystem.

Die Vorteile von 1-Wire-Netzen liegen klar auf der Hand: sehr geringe Hardwareanforderungen, niedrige Entwicklungs- und Produktionskosten und die (theoretische) Möglichkeit, über $2,8 \cdot 10^{14}$ verschiedene Chips der gleichen Familie anzusteuern. Diese Zahl leitet



Bild 5. Ein iButton als Ring.

DS2401 MEMORY MAP



030148 - 14

Bild 4. 64 Bit für eine eindeutige Identifikation.

sich aus der kompletten 64 Bit breiten Chip-ID ab (Bild 4), die im ROM (hier des DS2401) steckt.

Der Family-Code ist bei jedem 1-Wire Chip einer Gattung identisch. Dazu muss gesagt werden, dass der DS2401 nur eine ID enthält und nur gelesen, aber nicht geschrieben werden kann, was den 1-Wire Baustein zu einer Art Identifikationschip macht. Das höchstwertige Byte ist das Ergebnis eines CRCs (cyclic redundancy check), welcher sich auf die vorhergehenden sieben Bytes bezieht und eine fehlerhafte Datenübertragung in der Chip-ID aufdecken soll. Also geht die Einzigartigkeit eines Chips allein aus den mittleren 48 Bits aus – daher auch die Zahl von $2^{48} = 2,8 \cdot 10^{14}$ Möglichkeiten. Diese große Zahl von möglichen Busteilnehmern macht den DS2401 zu einer Art elektronischer Seriennummer zum Beispiel für Leiterplatten oder andere elektronische Geräte.

Diese einseitige Form der Kommunikation ist aber bei weitem nicht alles, was der 1-Wire-Bus zu bieten hat. Es gibt nämlich neben diesen reinen ID-Chips eine sehr große und stetig wachsende Anzahl von verschiedensten 1-Wire-Chips, angefangen vom einfachen EEPROM über komfortabel zu programmierende Echtzeituhren mit Eventcounter bis zum leistungsfähigen Mehrfach-Analog-Digitalwandler. Diese Chips weisen ebenfalls 2^{48} verschiedene Seriennummern auf, von denen jeweils nur eine einzige existiert. Diese Möglichkeit allein macht den 1-Wire-Bus schon zu etwas Besonderem, aber die Entwickler des Bus-Protokolls haben noch weitere, erstaunliche Merkmale und Funktionen implementiert. Der Busmaster kann zum Beispiel auch eigenständig nach allen mit dem Bus verbundenen Chip-IDs suchen (Search ROM) oder allen angeschlossenen Chips gleichzeitig einen Befehl schicken (Skip ROM). Diese Funktionen und

noch andere, starke Merkmale des 1-Wire Busses machen ihn zu einem sehr leistungsfähigen Datenbus, was sich auch an der großen Zahl der denkbaren Anwendungsmöglichkeiten widerspiegelt.

Wenn elektronische Geräte zunehmend „intelligenter“ werden und über viele Sensoren/Aktoren verfügen, stellt der 1-Wire-Bus das ideale System dar. Je mehr Slaves man jedoch in einem seriellen Bus anfindet, desto stärker wird er durch den Datenfluss elektrisch und informationstechnisch belastet. Dem kann man beim 1-Wire Bus relativ einfach aus dem Weg gehen, in dem man spezielle Busteilerbausteine (zum Beispiel der DS2409 in Verbindung mit dem DS2406) hinzufügt, die den Bus in mehrere kleinere Äste unterteilen, was der Busbelastung deutlich entgegenkommt. Ein Beispiel für einen gelungenen Einsatz des MicroLANs ist eine (öffentliche) Zugangskontrolle mit einem zentral installierten Controller-System. Durch die hohe Anzahl von möglichen IDs kann man viele Komponenten erfassen. Für den Einsatz in Zahlungssystemen und industriellen Zugangskontrollen hat der Entwickler des 1-Wire-Busses eine Art elektronische Metall-Tablette mit der Bezeichnung iButton entwickelt, die äußerst widerstandsfähig gegenüber äußeren Einflüssen ist (Bild 5). Diese kleinen, circa 16 mm durchmessenden ICs hält man zum Datenabgleich in ein kleines Terminal und schon ist der Zugang elektrisch registriert. Sie halten mühelos Kräfte im dreistelligen Newton-Bereich aus, was der Trittkraft eines Menschen entspricht – probieren Sie das mal mit einer Chipkarte (oder besser nicht)!

(030148)rg

Web-Adressen:

- www.dalsemi.com
- www.maxim-ic.com
- www.ibutton.com