

Sprach-/Datenkommunikation auf zwei Drähten und interne Bussysteme

Inhalt

1	Techniken	1
1.1	ISDN U _{p0}	2
1.2	ISDN U _{K0}	3
1.3	CAN	3
1.4	Serielle Schnittstelle	3
1.5	a/b	4
1.6	Powerline	4
2	Weitere Bussysteme für den internen Gebrauch	5
2.1	TWI	5
2.2	SPI	5
2.3	I ² S	5
3	Stromversorgung	6
4	Sicherung der Schnittstelle	6

Einige Anwendungen erfordern Verbindungen zu außenliegenden Sprechstellen, die deutlich weiter weg sind, als nur die erlaubten 90m einer Ethernetverbindung und außerdem keine andere Anbindung ans Netzwerk ermöglichen. Eine reine Verlängerung des Netzwerkes erfordert eine große Datenbandbreite, die nur mit hohem technischen Aufwand zu erreichen ist. Üblich für solche Anwendungen sind VDSL-Modems, die bis zu 1000m überbrücken können, wenn auch nur mit sehr stark verminderter Datenrate.

Für die reine Datenübertragung stehen mehr Techniken zur Verfügung, da die Anwendungsbreite hier deutlich höher ist. Für Daten entfällt gewöhnlich auch die Anforderung der Echtzeitübertragung. Oft fallen auch geringere Mengen von Daten an, als man sie für eine Audioübertragung unbedingt benötigt.

1 Techniken

Bei Neuinstallationen spielt es zumeist keine Rolle, welche Art von Kabel verlegt werden soll. Die preislichen Unterschiede sind vernachlässigbar gering und auch die Anzahl der einzelnen Adern in einem solchen Kabel spielen keine große Rolle.

Zweidrahttechniken sind interessant, weil man sie eventuell auf alten bestehenden Leitungen einsetzen kann (z.B. Telefonkabel) und weil sie teilweise über längere Strecken funktionieren. Oft ist es auch nur die Tatsache, dass man mit einem Drahtpaar leichter umgehen kann, als mit einem ganzen Kabelbaum.

Wir beschreiben hier nur die Zweidraht-Techniken, die wir derzeit in unserem Betriebssystem ellan:RTOS unterstützen und für die wir auch die Entwicklung entsprechender Hardware anbieten können. Es ist nicht ausgeschlossen, andere Konzepte zu realisieren, wenn dies aus besonderen Gründen notwendig sein sollte.

Für alle diese Schnittstellen bestehen bereits lizensierbare Stacks in unserem Betriebssystem, so dass die Entwicklung eines Transportprotokolls entfällt, wenn keine besonderen Anforderungen dies erfordern.

1.1 ISDN U_{p0}

U_{p0} ist die Zweidrahtvariante der üblichen ISDN-Schnittstelle S_0 . Man kann damit in vielen Fällen eine brauchbare Alternative zur aufwändigen Verlängerung eines Ethernets realisieren. Besonders für abgesetzte Sprechstellen eignet sich dieses Verfahren, wofür es auch entwickelt wurde.

Eine intelligente Reduzierung der notwendigen Datenbandbreite ermöglicht den Einsatz dieser bewährten Technik, die Leitungslängen sogar im Bereich von bis zu 4 Kilometern ermöglicht. Eine IP-Sprechstelle hat viele Daten zu verarbeiten. Sie muss nicht nur den Stream empfangen, den sie gerade wiedergibt, sondern auch alle Durchsagen, auch wenn sie aktuell nicht wiedergegeben werden müssen. Dazu kommt der Traffic, der durch SIP oder andere Protokolle verursacht wird und alle Systemmessages, die ein normales Computernetzwerk ständig erzeugt.

Die Lösung des Problems ist ein Terminal-Modus, bei dem die Sprechstelle nur noch den aktuell benötigten Audiostream und allenfalls Tastendrücke zu transportieren und interpretieren hat. Das spart auch Lizenzen, weil die Geräteintelligenz in den Adapter verlagert wird, im Endgerät also entfallen kann.

Eine U_{p0} -Schnittstelle kann direkt in das Endgerät integriert werden, ggfs. zusammen mit einer erneuerten Prozessorstruktur, die den Platz einspart, den man für die U_{p0} -Physik braucht. Aber auch als kleiner Adapter, der U_{p0} in Ethernet wandelt und umgekehrt, ist es realisierbar. Dieser wird dann an beiden Enden einer Leitung benötigt.

Bei Einsatz einer entsprechend leistungsfähigen CPU sind auch mehrkanalige Geräte (Konzentratoren) und Sicherheitsfunktionen möglich. Der Leistungsumfang einer Sprechstelle wird komplett im netzwerkseitigen Adapter abgebildet. Auch die Stromversorgung ist über die U_{p0} möglich, wobei eine Hin- und Rückkonvertierung von PoE möglich ist, falls gewünscht.

Die Hardware ist relativ günstig zu realisieren, auch unter Berücksichtigung der zusätzlichen Lizenzen. Der Entwicklungsaufwand ist abhängig von den Anforderungen allerdings relativ hoch und rechnet sich nur bei entsprechenden Stückzahlen. Wird die U_{p0} -Sprechstelle direkt im Endgerät realisiert, halbiert sich der Aufwand allerdings.

1.2 ISDN U_{K0}

U_{K0} ist die ISDN-Schnittstelle, die im öffentlichen Netz die Verbindung vom Hausanschluss zum Netzknoten herstellt. Datentechnisch sind sich U_{K0} und U_{p0} sehr ähnlich. Es gibt aber einige bedeutsame Unterschiede:

- Die U_{K0} hat eine Reichweite von bis zu 6km.
- U_{K0} benötigt unterschiedliche Hardware auf den beiden Seiten einer Verbindung, bei U_{p0} sind beide Seiten hardwaretechnisch gleich.
- Die notwendigen U_{K0}-Chips werden nicht mehr in Serie hergestellt und sind sehr teuer, wenn man sie überhaupt noch bekommt.

U_{K0} hat nur deshalb noch eine Bedeutung, weil sich noch sehr viele Geräte mit dieser Schnittstelle im Feld befinden.

1.3 CAN

Der CAN-Bus ist allgemein bekannt aus der Autoindustrie. Er wird benutzt um eine ziemlich große Menge von Sensoren und Aktoren miteinander zu verbinden. Dabei ist es möglich von einer üblichen Busstruktur abzuweichen und auch Stichleitungen in kleinerem Umfang zu ermöglichen. Auch Verkabelungen in Sternstruktur sind möglich.

Für unsere Anwendungen ist der CAN-Bus deshalb so interessant, weil er bis zu 500m reicht, wenn auch nur mit ca. 125kbps, also weniger als bei ISDN. Dafür handelt es sich hier aber um einen Feldbus, der es ermöglicht, mehrere Endstellen zu betreiben.

Wegen der hohen Verbreitung, sind Peripheriebausteine für den CAN-Bus in einer großen Vielfalt verfügbar und preiswert. Das kann ein Grund sein, um dieses Interface einzusetzen. Auf kurzen Strecken sind Geschwindigkeiten von bis zu 1mbps möglich.

Ein CAN-Bus kann mit nur zwei Drähten realisiert werden, es gibt aber auch Realisierungen mit mehr Drähten, die physikalische Vorteile bieten.

1.4 Serielle Schnittstelle

Serielle Schnittstellen gibt es in sehr vielen Ausführungen. Sie unterscheiden sich in Signalpegeln, Geschwindigkeit, Datenformat und anderen physikalischen Eigenheiten. Aus der Sicht des Prozessors, werden serielle Schnittstellen logisch über einen UART (Universal Asynchronous Receiver Transmitter) realisiert. Mit geschickter Programmierung kann man eine serielle Schnittstelle aber auch mit gewöhnlichen digitalen Ein- und Ausgängen eines Prozessors realisieren.

Es gibt auch synchrone serielle Interfaces, aber die werden nicht mit zwei Drähten realisiert. Die asynchrone serielle Schnittstelle ohne zusätzliche Leitungen wird oft serielle Zweidrahtschnittstelle genannt. Das ist aber eigentlich nicht richtig, denn neben den beiden Leitungen für die Sende- und Empfangsrichtung, benötigt man noch Zugriff auf eine gemeinsame Masse.

Ein echter Zweidraht ist es also nur, wenn man die Masse nicht separat weiterleiten muss (z.B. geräteintern), oder wenn man Daten in nur eine Richtung bewegt.

Allerdings gibt es physikalische Ausführungen, die auch auf längeren Strecken mit wirklich nur zwei Drähten auskommen. Die bekanntesten sind wohl EIA-485 und EIA-422 (früher RS485 und RS422). Hier behilft man sich damit, dass die Daten in einem PingPong-Verfahren ausgetauscht werden, also wechselweise mal in die eine, dann in die andere Richtung. Es ist sogar möglich eine Busstruktur zu realisieren.

Für serielle Schnittstellen gibt es kein allgemeingültiges Datenprotokoll. Je nach Anwendung muss man ein eigenes erfinden oder etwas übernehmen, das schon anderswo erfolgreich eingesetzt wird.

Serielle Schnittstellen können von uns in vielen Formen realisiert werden, was nicht nur für die Protokolle, sondern auch für die Leitungsphysik gilt. Stromschleifen oder optische Anbindungen sind ebenfalls möglich.

Serielle Leitungen übertragen die Informationen im Basisband, also genau mit der Geschwindigkeit, die von der Frequenz des Nutzsignals vorgegeben wird. Je höher die Frequenz, also die Basisbandbreite, desto anspruchsvoller die physikalische Ausführung der Schnittstelle.

1.5 a/b

Um gewöhnliche analoge Telefone an ein IP-System anzubinden, wird ein a/b-Wandler benötigt. Dieser Wandler muss sich zum Netzwerk hin wie eine normale IP-Sprechstelle verhalten. Einige Leistungsmerkmale bleiben einem analogen Gerät zumeist verwehrt. Vor allem das automatische Annehmen von Verbindungen ist normalerweise nicht möglich (Ausnahmen sind realisierbar).

Der Wandler muss Schleifenstrom liefern, das Abheben auswerten und auch die Bedienung der vorhandenen Elemente. Gewöhnlich sind das DTMF und Erdtaste, aber auch Pulswahl ist möglich.

Zu berücksichtigen sind u.U. auch Fähigkeiten zur Rufnummernanzeige (V.23 als Übertragungsverfahren der Information) oder andere analoge Datenübertragungsverfahren (Modem). Der Wandler kann auch für lange analoge Verbindungen realisiert werden.

Wir haben auch schon Modems selbst programmiert. Damit sind geringe Datenraten auf langen Strecken realisierbar. Meistens geht es aber um die Herstellung von Kompatibilität zu bestehenden Systemen.

1.6 Powerline

Powerline bietet die Möglichkeit der Übertragung von Daten über die Netzstromleitung. Was auf den ersten Blick wie ein Geniestreich aussieht, ist andererseits aber auch mit Nachteilen behaftet. Das Hauptproblem liegt in der Datensicherheit, denn ob Powerline eine stabile Verbindung bietet, liegt in erster Linie an der Infrastruktur, also der Leitungsführung. In manchen Umgebungen funktioniert es problemlos, in anderen überhaupt nicht. Oft kommt es zu Aussetzern, je nach Last auf der Stromleitung. Es gibt Fälle, in denen die Übertragung abbricht, wenn ein Kühlschrank anspringt.

Trotzdem kann Powerline eine Alternative zu WLAN oder dem Verlegen neuer Kabel sein. Allerdings ist die notwendige Hardware im Gerät (es handelt sich um eine kleine zusätzliche Leiterplatte) nicht eben preiswert.

Die Powerline betreiben wir gewöhnlich einfach mit TCP/IP, wie auch alle anderen Netzwerkanschlüsse.

2 Weitere Bussysteme für den internen Gebrauch

Für die Verbindungen innerhalb eines Gerätes können die oben genannten Zweidraht-Interfaces ebenfalls eingesetzt werden. Allerdings ist der technische Aufwand dafür meistens einfach zu hoch. Allenfalls der CAN-Bus kann hier sinnvoll sein.

Einfache Aufgaben können direkt mit Ein- und Ausgängen des Prozessors erledigt werden, es gibt allerdings auch spezielle Interfaces, die wir gern für die geräteinterne Kommunikation einsetzen.

2.1 TWI

Hier ist der Name Programm (TWI = Two Wire Interface), aber bekannter ist diese Schnittstelle unter dem Namen I²C. Dieser Name wurde aber von einem Hersteller geschützt, weshalb man synonym von TWI sprechen kann.

TWI ist ein Bus, an dem mehrere Endgeräte gleichzeitig existieren können. Diese haben jeweils eine feste Adresse, so dass sie von der Hostseite aus gezielt angesprochen werden können. Der Host ist in diesem Zusammenhang der Hauptprozessor des Gerätes,

Die möglichen Geschwindigkeiten sind hoch genug für alle möglichen Anwendungen. In normaler Ausführung sind 0,1 bis 1,0mbps möglich. Es sind auch spezielle schnellere Implementationen bekannt, für die es aber meistens bessere Alternativen gibt.

TWI eignet sich für den Anschluss von Displays, I/O-Bausteinen, Flash-Speicher usw. Wir verwenden diese Schnittstelle gern für Erweiterungen, die optional angeschlossen werden können.

2.2 SPI

SPI ist dem TWI von der Anwendung her sehr ähnlich, es gibt aber einige Unterschiede:

- Es werden drei anstatt nur zwei Drähte benötigt.
- Die Schnittstelle überträgt in beide Richtungen gleichzeitig (voll duplex).
- SPI ist deutlich schneller.

Wenn die Geschwindigkeit keine große Rolle spielt, bestimmt die Anwendung die vorzuziehende Schnittstelle. Man muss ermitteln, welche Bausteine für welche Aufgabe verfügbar sind. Bei gleichen Möglichkeiten gibt man SPI den Vorrang.

SPI ist auch für eine schnelle Kommunikation zwischen zwei Prozessoren geeignet. Das könnte z.B. eine Verbindung zwischen einem Echtzeitprozessor und einem grafischen Subsystem (Linux o.ä.) sein.

2.3 I²S

Obwohl es ähnlich aussieht wie I²C, ist I²S (Inter IC Audio) etwas völlig anderes. Diese Schnittstelle ist speziell für die Übertragung digitaler Audiodaten geschaffen worden. I²S ist auch kein Bus, sondern verbindet lediglich zwei beteiligte Endpunkte. Wegen der synchronen Übertragung, werden für I²S drei Drähte benötigt.

Diese Schnittstelle verwenden wir, um geräteintern Audioeinheiten an die CPU anzubinden. Nach außen verwendet man andere Interfaces, die hier nicht weiter besprochen werden sollen.

3 Stromversorgung

Eine wichtige Eigenschaft jeder Schnittstelle, zumindest wenn sie längere Strecken überbrücken soll, ist ihre Fähigkeit auch eine Stromversorgung zur Verfügung zu stellen. Natürlich besteht immer die Möglichkeit über parallele Leitungen Strom zu liefern. Eine Zweidrahtverbindung ist dann aber keine echte Zweidrahtverbindung mehr.

Die beiden bereits genannten ISDN-Schnittstellen können gleichzeitig eine Stromversorgung realisieren. Diese ist ausreichend für kleinere Verbräuche und damit auch für ganze Endgeräte, die nicht allzu energiehungrig sind (Sprechstellen, Telefone). Das funktioniert ähnlich wie bei PoE (Power over Ethernet), nur dass dort deutlich mehr Strom geliefert werden kann.

Serielle Schnittstellen können ebenfalls so ausgeführt werden, dass eine Versorgungsspannung transportiert wird. Die physikalische Ausführung muss es nur hergeben.

Stromversorgung über a/b ist in begrenzter Form ebenfalls üblich. Man kennt das von Telefonleitungen aber auch in der Studioteknik, wo z.B. Mikrofone über eine Phantomspeisung versorgt werden.

Powerline bringt den Strom sozusagen von zuhause mit, da hier die Daten über den Netzstrom geleitet werden, nicht umgekehrt, wie in den anderen Fällen. Deshalb gibt es hier auch so gut wie keine Einschränkung in der Leistung.

4 Sicherung der Schnittstelle

Egal, welche Schnittstelle man auch benutzt, die übertragenen Daten sind ungesichert und können von Jedermann abgehört werden, wenn er Zugriff auf die Leitungsführung hat. Es ist nicht die Aufgabe einer Schnittstelle, Inhalte zu verschleiern oder zu verschlüsseln.

Will man die Daten sichern und für Unbefugte unbrauchbar machen, braucht man ein zusätzliches Verschlüsselungsverfahren. Das funktioniert mit allen digitalen Schnittstellen, fordert aber je nach Datenbandbreite mehr oder weniger Rechenleistung vom System.

Um verschlüsselte Daten zu transportieren, empfiehlt sich ein gesichertes Übertragungsprotokoll mit den entsprechenden Algorithmen. Wir können solche Lösungen in vielen Formen anbieten.

Für weitere Informationen stehen wir Ihnen jederzeit zur Verfügung.