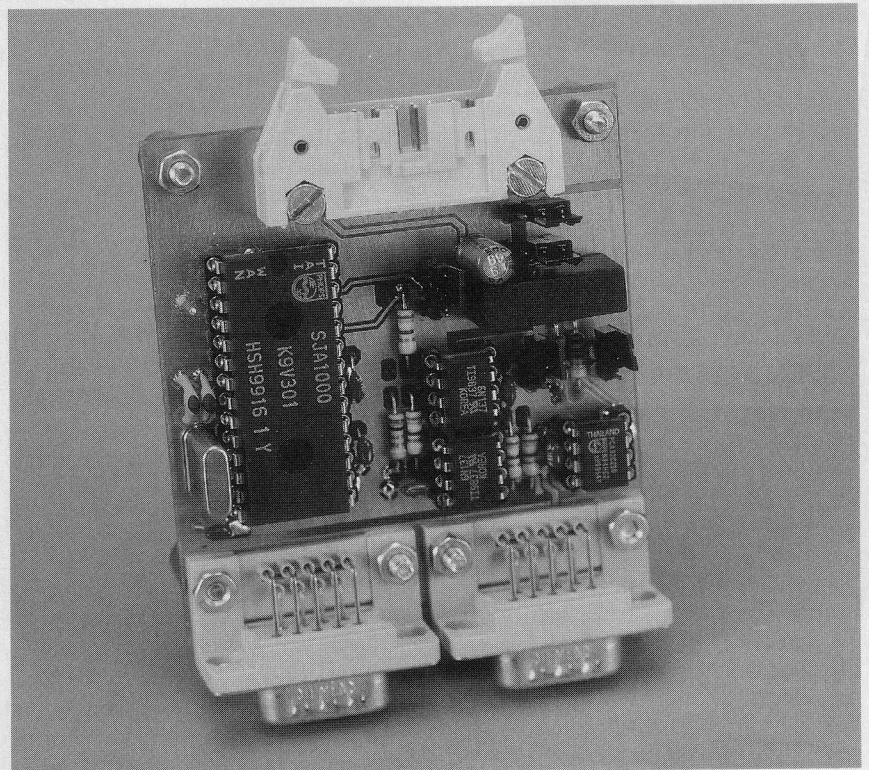


Der CAN – Bus

Intelligente, dezentrale Datenkommunikation für den Praktiker (Teil 3)

Wie im vorhergehenden Teil abschließend versprochen, beginnen wir nun mit dem praktischen Aufbau eines CAN-Bussystems. Dazu wird zunächst eine CAN-Bus-Interface-Baugruppe vorgestellt, die an alle gängigen Mikrocontroller-Systeme angeschlossen werden kann und diese somit bei Bedarf CAN-tauglich macht.



Grundsätzlich haben heutige CAN-Ankopplungen den in **Bild 1** dargestellten 3-Chip-Aufbau. Der Mikrocontroller schreibt dabei lediglich die zu übertragenden Nutzdatenbytes (0 bis 8 Stück) in den CAN-Protokoll-Chip, füllt das Identifier-Feld und das DLC-Feld aus und setzt das RTR-Bit entsprechend. Den Rest erledigt der CAN-Controller ganz alleine:

- Berechnung der CRC-Checksumme
- Hinzufügung der restlichen Felder
- Zugriff auf den Bus
- Aussenden der Daten
- Fehlererkennung und -behandlung
- etc.

Der Mikrocontroller erhält dann als Rückmeldung entweder eine Bestätigung über das erfolgreiche Aussenden der Daten oder aber eine entspre-

chende Fehlermeldung und kann dann darauf geeignet reagieren.

Das entsprechende geschieht in Empfangsrichtung: der CAN-Chip empfängt die CAN-Frames, überprüft selbständig die CRC-Checksummen, trennt alle überflüssigen Felder des Frames ab und stellt dem Mikrocontroller entweder die korrekt empfangenen Daten oder einen Fehlerhinweis zur Verfügung. Ergänzt wird der CAN-Protokoll-Chip noch durch den CAN-Transceiver-Baustein, der für die direkte Busanpassung verantwortlich ist. Für den Anwender bleibt so der für die CAN-Bus-Anbindung seiner Applikation erforderliche Hardware- und Software-Aufwand sehr gering. Mittlerweile gibt es auch schon komplette Mikrocontroller, die den CAN-Controller als zusätzliche Funktionsbaugruppe bereits auf ihrem Chip enthalten, so

daß aus der 3-Chip- eine 2-Chip-Lösung wird. Bevor wir Ihnen nun die Schaltung des CAN-Bus-Interfaces vorstellen, sind noch einige Informationen vorab erforderlich.

DIE AKZEPTANZ-FILTERUNG

Sie haben gesehen, daß in einem CAN-System (mit Standard-Frame-Format) bis zu 2048 unterschiedliche Nachrichtenobjekte ausgetauscht werden können. Nun ist es sicherlich nicht für jede angeschlossene Station notwendig, alle Data/Remote-Frames zu empfangen: für die Station K sind z.B. nur die Frames mit den Identifier 129,1345 und 1999 wichtig, alle anderen 2045 möglichen Frames sind für K uninteressant. Damit K nun nicht jeden Frame erst einmal empfangen und der Mikrocontroller den Identifier auf "passend/nicht passend" überprüfen muß (was sehr zeitaufwendig ist), wäre eine Vorfilterung der Identifier sicherlich sinnvoll, die nur diejenigen Frames an den Mikrocontroller weitergibt, die auch für diese Station wichtig sind.

Eine solche Vorauswahl der empfangenen Frames nennt man Akzeptanzfilterung. Mit anderen Worten: Sie können den CAN-Chip so programmieren, daß er nur Frames mit bestimmten Identifier zum Mikrocontroller durchläßt. Alle anderen Frames werden zwar empfangen und überprüft (inkl. Fehlerbehandlung), aber nicht weitergegeben. Der Mikrocontroller wird hier also von erheblichen Vergleichsaufgaben entlastet und kann sich so voll auf die Weiterverarbeitung der empfangenen, wirklich für ihn bestimmten, Data/Remote-Frames konzentrieren.

Bei der Realisierung dieser Akzeptanzfilterung gibt es zwei (leicht) unterschiedliche Konzepte:

BasicCAN-Chips

Diese Bausteine besitzen nur ein einfaches Filter (z.B. nur 8 Bit breit) und führen daher nur eine recht grobe "Vorfilterung" durch, d.h. sie lassen immer ganze Gruppen von Identifier zum Mikrocontroller durch, z.B. alle Identifier im Bereich von 700 ... 707 gelangen durch das Filter. Eine Herausfilterung von nur exakt einem einzigen Identifier ist nicht möglich.

Hier muß dann der Mikrocontroller unter Umständen selber noch einen Teil der Auswahlfilterung durchführen, um letztendlich den gewünschten Frame zu erhalten. Weiterhin werden auch Remote-Frames, die für diese Station bestimmt und durch das Filter gelangt sind, an den Mikrocontroller weitergeleitet, der dann erst einmal die Antwortdaten zusammenstellen und an den CAN-

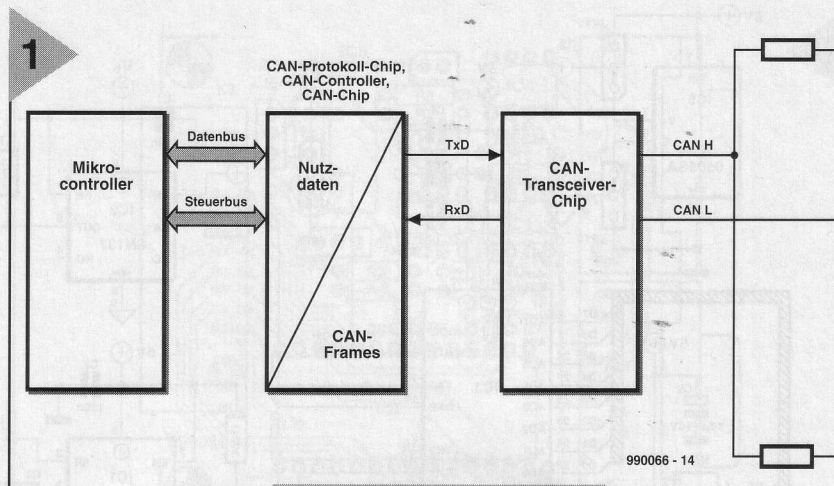


Bild 1. Blockschaltbild der 3-Chip-CAN-Lösung.

Controller übergeben muß.

FullCAN-Chips

Mit diesen CAN-Controllern kann man eine programmierbare, exakte Herausfilterung von Identifier vornehmen, d.h. man kann den FullCAN-Chip so einstellen, daß er nur die Frames weitergibt, die einen bestimmten Identifier haben, z.B. nur die Frames mit dem Identifier 798. Sehr oft ist es auch möglich, mehrere Identifier anzugeben, z.B. bis zu 16 Stück, die dann ganz individuell aus den Datenströmen "herausgefischt" werden. Der Nachteil dabei ist aber, daß eine größere Anzahl von Frames oder Frames mit anderen Identifier nicht weitergegeben werden, es sein denn, man programmiert den Controller perma-

nent um.

Sollen also viele Frames mit unterschiedlichsten Identifier empfangen werden können, so ist der Einsatz eines BasicCAN-Chips sinnvoller, wobei dann der Mikrocontroller aber einen Großteil der Filterarbeit erledigen muß, ein entsprechend leistungsfähiger Mikrocontroller also erforderlich wird.

Ein weiterer Vorteil bei FullCAN-Controllern ist die Tatsache, daß der Mikrocontroller die Antwort auf einen Remote-Frame in den CAN-Chip einprogrammieren kann. Empfängt der CAN-Controller nun einen gültigen (zulässigen) Remote-Frame für diese Station, so kann er sofort selbständig den Antwort-Data-Frame senden, ohne den Mikrocontroller damit belästigen zu müssen.

Tabelle 1: Technische Daten

CAN-Controller-Chip: SJA 1000 von Philips

▶ Mikrocontroller-Interface umschaltbar für Intel- oder Motorola-Mikrocontroller o.ä.

▶ Betriebsmodus 1: PCA82C200 pin-, hardware- und software-kompatibel

- CAN2.0A und CAN2.0B passiv,
- Standard-Frame-Format,
- Datenübertragungsraten bis zu 1 MBit/s,
- einfaches Akzeptanzfilter, BasicCAN.

▶ Betriebsmodus 2: "PeliCAN-Modus"

- Extended- und Standard-Frame-Format,
- Datenübertragungsraten bis 1 MBit/s
- vollständig CAN2.0B-fähig,
- erweitertes Akzeptanzfilter, jedoch nur BasicCAN-Eigenschaft,
- weitere, interessante Zusatzfunktionen.

CAN-Transceiver-Chip: PCA82C250 von Philips

▶ Kompatibel zu ISO/DIS11898, High-Speed-CAN,

▶ Datenübertragungsraten bis 1 MBit/s,

▶ interne Schutzbeschaltungen gegen Störungen speziell im Automobil-Bereich,

▶ interne Schutzbeschaltungen gegenüber Kurzschluß und thermische Überlast,

▶ ein nicht mit Betriebsspannung versorgter Knoten stört den Bus nicht,

▶ mit diesem Transceiver-Chip können CAN-Systeme mit bis zu 110 Knoten (Stationen) aufgebaut werden.

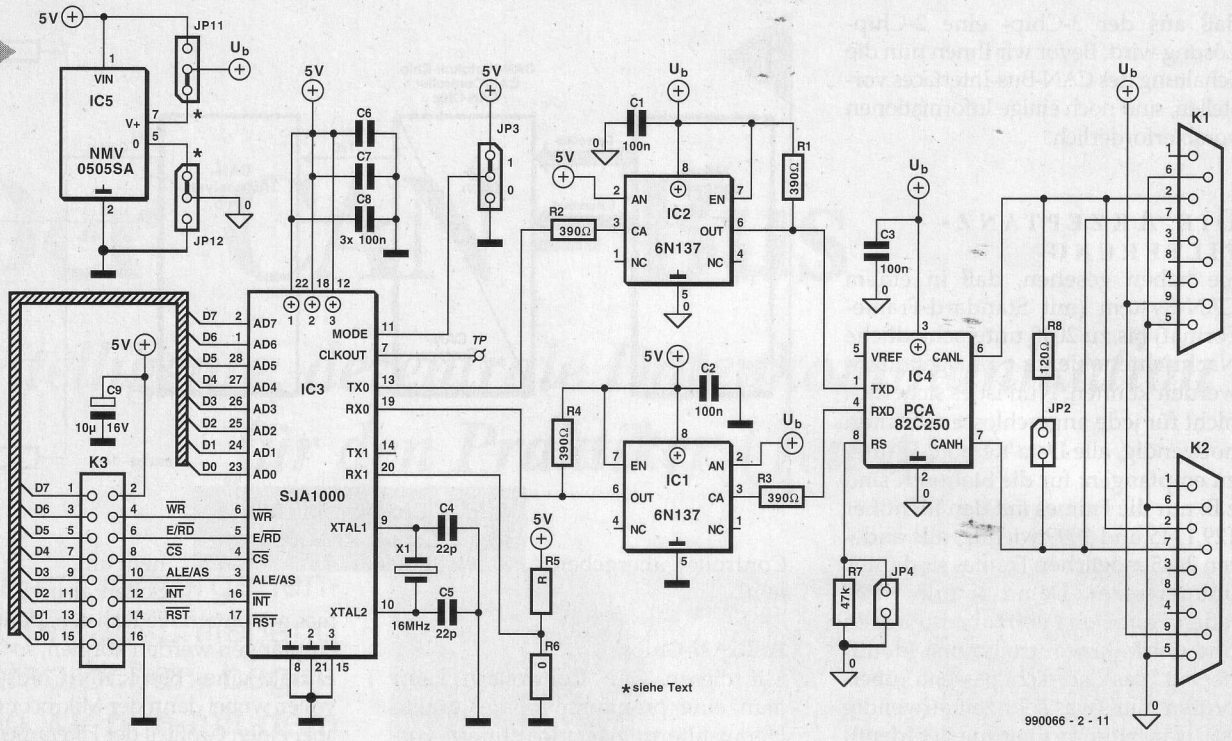


Bild 2. Schaltplan des CAN-Bus-Interfaces, das aus einem CAN-Controller, einem CAN-Transceiver, einem DC/DC-Konverter und zwei Optokopplern besteht.

Beim heutigen Stand der Technik verweisen die Grenzen zwischen Basic-CAN und FullCAN immer mehr (Full-CAN-Chips werden immer leistungsfähiger, d.h. sie können immer mehr einzelne Identifier herausfiltern und die Datensätze abspeichern) und bei vielen CAN-Chips kann man sogar softwaremäßig zwischen beiden Betriebsarten umschalten.

VERSION 2.0A/2.0B-KOMPATIBILITÄT

Wie Sie bei der Beschreibung der Frame-Formate gesehen haben, gibt es das Standard-Frame-Format mit 11-Bit Identifier und das Extended-Frame-Format mit 29-Bit Identifier. Hier muß man besonders aufpassen, welche CAN-Controller man verwendet, wenn man beide Frame-Formate in einem Bussystem mischen will (was durchaus möglich und auch zulässig ist):

Controller mit 2.0A-Fähigkeit

Diese Chips können nur Standard-Frames verarbeiten und melden beim Empfang eines Extended-Frames immer einen Fehler, d.h. sie senden dann immer einen Error-Frame auf den Bus und die Kommunikation kommt dadurch völlig zum Erliegen. Solche Bausteine dürfen nur in Systemen mit Standard-Frames verwendet werden.

Controller mit 2.0A-Fähigkeit und 2.0B Passiv-Eigenschaft

Diese Bausteine tolerieren Extended-Frames mit 29-Bit Identifier, d.h. sie

empfangen die Extended-Frames, führen eine Fehlerprüfung durch und antworten auch entsprechend entweder mit einem ACK-Bit oder mit einem Error-Frame.

Die Kommunikation wird durch diese Bausteine also nicht gestört. Allerdings werden die empfangenen Extended-Frame-Daten nicht gespeichert oder weitergeleitet, weil diese Bausteine nun einmal nur auf die Weiterverarbeitung von Frames im Standard-Format ausgelegt sind. Solche Bausteine können also problemlos in gemischten Systemen eingesetzt werden.

Controller mit 2.0B-Fähigkeit

Diese Chips verarbeiten (speichern und weiterleiten) problemlos Standard- und Extended-Frames, sie sind also universell einsetzbar.

Wenn Sie sich nun für einen CAN-Controller oder einen Mikrocontroller mit ON-Chip-CAN-Baugruppe entscheiden müssen, so haben Sie die Qual der Wahl, denn fast jeder große Halbleiterhersteller hat solche Bausteine im Sortiment. Um den sich "täglich" ändernden, aktuellen Chip-Stand zu erfragen, ist ein Besuch auf den Internet-Homepage folgender Hersteller sicherlich angebracht, zumal man von dort auch sofort die entsprechenden Datenblätter herunterladen kann: Philips, Siemens, Intel, Hitachi, Motorola, NSC, SGS, Temic, Texas Instrument etc. (das ist hier nur eine Auswahl!).

DAS CAN-BUS-INTERFACE

Nach so viel an Grundlagen und Theorie kommen wir nun endlich zur Schaltung des kompakten, leistungsfähigen CAN-Bus-Interfaces, zu dessen Beschreibung erfrischend wenig Text erforderlich ist.

Die Schaltung des CAN-Bus-Interfaces ist in Bild 2 zu sehen, den Bestückungsplan und die Layouts der doppelseitigen, durchkontaktierten Platine findet man in Bild 3.

Die Daten dieser Buskoppelstufe zeigt die Tabelle 1. Der als CAN-Controller verwendete SJA 1000 (Innenleben siehe Bild 4) ist der Nachfolger des legendären CAN-Chips PCA82C200 und mit diesem im Betriebsmodus 1 pin-, hardware- und softwarekompatibel. Die Schnittstelle zum Mikrocontroller hin ist für Intel- oder Motorola-Mikrocontroller umschaltbar. Als CAN-Transceiver kommt der PCA82C250 (ebenfalls von Philips) zum Einsatz. Die Anschluß des Mikrocontroller-Systems an das CAN-Bus-Interface erfolgt über ein Flachbandkabel von maximal 10 cm Länge an den 16-poligen Wannenstecker K3. Über diese Schnittstelle steuert der Mikrocontroller den CAN-Controller (IC3) an: es werden Daten-, Steuer- und Zustandinformationen ausgetauscht. Diese Werte bereitet der SJA1000 dann "CAN-mäßig" sowohl in Sende- als auch in Empfangsrichtung auf. Für den Mikrocontroller sieht der CAN-