Feldbussystem - CAN

CAN - Controller Area Network

Versuchsbetreuer
Dipl.-Ing. Th. Wegner
Raum: W1314
Tel.: 498 7267

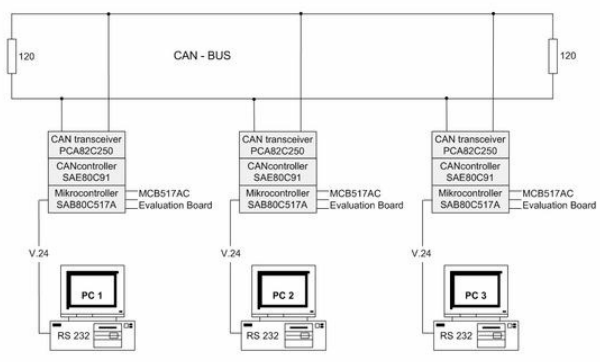
PRT-Labor
Raum: W1216
Tel.: 498 7256

1. Versuchsziel

Es sollen die Funktion und die Programmierung des Mikrocontrollers SAB 80C517A und des Full-CAN Controllers SAE 81C90 am Beispiel der seriellen Datenkommunikation über den CAN-Bus demonstriert werden.

2. Grundlagen

Das serielle Feldbussystem CAN (Controller Area Network) wurde ursprünglich für den Einsatz in der Kraftfahrzeugtechnik konzipiert und fand durch seine Echtzeit- und Multimasterfähigkeit eine weitere Verbreitung in der industriellen Automatistierungstechnik, der Medizin- und Verkehrstechnik und der Gebäudeautomation. Zur Protokollanalyse ist im Labor ein Controller Area Network bestehend aus 3 Knoten, die über eine verdrillte Zweidrahtleitung verbunden sind, installiert. Als Knotenrechner dient der SAB 80C517A, die ROM-lose Version des Mikrocontrollers SAB 80C517 aus der Familie der 8051-Prozessoren von Infineon, auf dem Mikrocontrollerentwicklungsboard MCB-517AC. Die Entwicklungsboards sind jeweils über eine serielle Schnittstelle mit einem PC verbunden. Der EPROM-residente Monitor des MCB-517AC gestattet eine Programmtestung unter Echtzeitbedingungen. Das CAN-Protokoll ist inzwischen von mehreren Halbleiterherstellern in Feldbusprozessoren implementiert. Im Versuchsaufbau kommt der Full-CAN-Controller SAE 81C90 von Infineon zum Einsatz. Die Kopplung zwischen Mikrocontroller und CAN-Controller im Knotenrechner erfolgt über das CPU-Interface, so dass der Dual-Port-RAM des SAE 81C90 ab Adresse 0F700h des externen CPU-Datenspeichers erscheint. Als CAN-Transciever für die physikalische Busankopplung dienen ISO/DIS-11898-kompatible integrierte Treiberchips, die eine Übertragungsrate bis 1 MBit/s bei einer Buslänge von 40 m ermöglichen. Die Programmentwicklung erfolgt unter KIEL-µVision in den Programmiersprachen "C51" oder Assembler. Das Datenpaket des CAN-Protokolls setzt sich aus 7 verschiedenen Bit-Feldern zusammen, aus dem Start of Frame, dem Arbitration Field, dem Control Field, dem Data Field, dem CRC Field, dem Acknowledge Field und dem End of Frame.



Blockschaltbild des Versuchsaufbaus

Die CAN-Protokollcontroller realisieren hardwaremäßig folgende Aufgaben:

- Erzeugung des CAN-Protokollframes
- Serialisierung der zu sendenen Daten
- Einfügen bzw. Entfernen von Stuff-Bits
- Berechnung bzw. Überprüfung der CRC-Prüfsequenz
- Busarbitrierung
- Fehlererkennung- und Signalisierung
- Erzeugen bzw. Überprüfen des ACK-Bits
- Synchronisation des empfangenen Bitstroms
- Assemblierung der empfangenen Daten
- Nachrichtenverwaltung (Akzeptanzfilterung, Speicherung)

Das CAN-Protokoll besitzt die folgenden Eigenenschaften:

- Priorisierung der Datenframes: Ein Identifier definiert eine statische Proirität während des Buszugriffs.
- geringe Verzögerungszeiten bei der Datenübertragung: Buszugriff nach CSMA/CA, kurze Frames
- flexible Konfiguration: hinzufügen von Stationen ohne Änderungen an Hard- und Software
- zeitsynchrone Multicastfähigkeit: jeder Knoten kann Nachricht gleichzeitig empfangen
- systemweite Datenkonsistenz durch Multicasting und Fehlersignalisierungsmechanismen
- Multi Master System: Buszuteilung durch verlustlose Busarbitrierung
- Fehlererkennung- und Signalisierung: Überwachung der Frames und der Stationen im CAN-Bus

3. Studienfragen

- 3.1 Nennen Sie verschiedene Feldbussysteme und deren Einsatzgebiete.
- 3.2 Erläutern Sie die wichtigsten Merkmale des CAN-Systems und deren Vorteile für den Einsatz auf der Sensor/Aktuator-Ebene.
- 3.3 Welche Frames werden im CAN-Protokoll unterschieden?
- 3.4 Erläutern Sie den Aufbau eines Datenframes im CAN-Protokoll.
- 3.5 Wie erfolgt die zerstörungsfreie Busarbitrierung mit Hilfe der Nachrichtenidentifier?
- 3.6 Erläutern Sie den Aufbau der RegisterMap des CANcontrollers SAE 81C90.
- 3.7 Welche Steuerregister werden für Initialisierung des CANcontrollers benötigt?
- 3.8 Wie erfolgt das Anlegen von Kommunikationsobjekten?
- 3.9 Wie ist der externe CANcontroller in das Interruptsystem des Mikrocontrollers eingebunden?

4. Aufgaben

- 4.1 Auswertung der Studienfragen.
- 4.2 Berechnen Sie die Initialisierungswerte der Register BL1, BL2 und BRPR für eine Baud-Rate von 100 kBit/s und 1 MBit/s bei einem ControllerTakt von 16 MHz.
- 4.3 Legen Sie im Programm für jeden CANknoten ein Kommunikationsobjekt an.
- 4.4 Vervollständigen Sie das Programm für den Sendeknoten. Wodurch wird das Senden einer Nachricht auf den Bus gestartet?
- 4.5 Realisieren Sie die Datenübertragung von einem Sender auf zwei Empfänger. Wie kann festgestellt werden, ob eine neue Nachricht empfangen wurde?
- 4.6 Realisieren Sie den unabhängigen Duplexverkehr zwischen den CANknoten.
- 4.7 Programmieren Sie eine Datenübertragung über Remote Frames.

5. Literatur

- CAN Spezifikation Version 2.0, Robert Bosch GmbH Stuttgard 1991
- Internationale Standardisierung: ISO-DIS 11898, ISO-DIS 11419-1, CiA Draft Standard 102/2.0
- Konrad Etschberger, Controller Area Network, Carl Hanser Verlag München Wien, 1994
- Elektronik plus Automatisierungspraxis 1, Sonderheft Feldbusse, 1992
- Elektronik plus Automatisierungspraxis, Sonderheft CAN, 6/1993

MCB517AC_Evaluation_Board.pdf	384 K
CAN-Controller__SAE_81c90.pdf	519 K
CAN_Physical_Layer__Specifikation.pdf	151 K
CAN_Physical_Layer_intergrated_Transceiver_PCA82C250.pdf	101 K
Connecting_C166_and_C500_Mikrocontroller_to_CAN.pdf	526 K
Mikrocontroller_mit_CAN-Controller_Siemens_C167CR.pdf	4.2 M
Mikrocontroller_mit_CAN-Controller_Siemens_C515C.pdf	2.4 M

Links - Feldbusse

- The international trade association CAN in Automation - CiA
- ASI: Aktor-Sensor-Interface
- BitBus
- CAN: Controller Area Network
- DIN-Meßbus
- EIB: European Installation Bus
- InterBus-S
- LON: Local Operating Network
- ProfiBus
- FlexRay
- CAN-Monitor mit CANUSB Interfaceadapters

Links - Funknetze

- Bluetooth
- nanoNET
- ZigBee / IEEE 802.15.4

6. Anhang

CAN Frames

- Data Frame
- Remote Frame
- Error Detection
- Error Confinement

CAN Data Frame

Mit einem Data Frame können im CAN-Protokoll bis zu 64 Bit Daten von einem Sender zu einem oder mehreren Empfängern (Broadcasting) übertragen werden.

Bit	Bedeutung	Beschreibung
		Interframe Space mindestens 11 Bit lang.
0	SOF-Bit	Das dominante Start of Frame Bit kennzeichnet den Beginn eines Daten- oder Datenanforderungsframes. Das SOF-Bit darf durch den CAN-Busknoten nur gesendet werden, wenn der Bus mindestens 11 Bitzeiten im Ruhezustand (rezessiver Pegel) war. Alle Busteilnehmer synchronisieren sich auf die Flanke des SOF-Bits.
1-11	ID10-ID0	11 Bit Identifier (2048) zur Kennzeichnung des Nachrichtenobjektes und dessen Priorität. MSB-Bit wird zuerst übertragen.
12	RTR-Bit	Durch das Remote Transmission Request Bit kann zwischen Daten- und Datenanforderungsframe unterschieden werden. Das RTR-Bit eines Datenframes ist dominant.
13,14	r1, r0	reservierte Bits
15-18	DLC3-DLC0	Durch den Data Length Code wird die Byte-Anzahl (0-8) des nachfolgenden Datenfeldes angegeben
19-63	Datafield	Das Datenfeld enthält die 0 bis 8 Byte Nutzerinformation des CAN-Frames.
64-78	CRC	Enthält die 15 Bit Prüfsequenz über das SOF-Bit bis zum letzten Bit des Datenfeldes.
79	CRC-End	rezessives CRC-Begrenzungsbit (cyclic redundancy code)
80	ACK-Slot	Acknowledge Slot für die stationsneutrale posititve Empfangsbestätigung des CAN-Protokolls ist von zwei rezessiven Bits eingerahmt. Der Sender legt ein rezessives Bit an und erwartet die Empfangsbestätigung in Form eines domianten Pegels durch mindestens einen Empfänger.
81	ACK-End	Acknowledge Delimiter, rezessives Begrenzungsbit
82-88	EOF-Field	Jeder Daten- und Datenanforderungsframe wird durch 7 rezessive End of Frame Bits beendet. Mit dem ACK-End Bit beenden 8 rezessive Bits die Nachricht.
89-91	Intermission	ACK-End + EOF-Field + Intermission Field ergibt mindestens 11 rezessive Bits InterFrame Space.

CAN Remote Frame

Mit einem Remote Frame können Empfänger die Übertragung von Daten mit dem gleichen Identifier bei einem Sender anfordern. Ein Remote Frame (RTR=1) enthält kein Datenfeld.

Bit	Bedeutung	Beschreibung
		Interframe Space mindestens 11 Bit lang.
0	SOF-Bit	Das dominante Start of Frame Bit kennzeichnet den Beginn eines Daten- oder Datenanforderungsframes. Das SOF-Bit darf durch den CAN-Busknoten nur gesendet werden, wenn der Bus mindestens 11 Bitzeiten im Ruhezustand (rezessiver Pegel) war. Alle Busteilnehmer synchronisieren sich auf die Flanke des SOF-Bits.
1-11	ID10-ID0	11 Bit Identifier (2048) zur Kennzeichnung des Nachrichtenobjektes und dessen Priorität. MSB-Bit wird zuerst übertragen.
12	RTR-Bit	Durch das Remote Transmission Request Bit kann zwischen Daten- und Datenanforderungs- frame unterschieden werden. Das RTR-Bit eines Remoteframes ist rezessiv.
13,14	r1, r0	reservierte Bits
15-18	DLC3-DLC0	Durch den Data Length Code wird die Größe des angeforderten Datenfeldes angegeben.
19-33	CRC	Enthält die 15 Bit Prüfsequenz über das SOF-Bit bis zum letzten Bit des DLC-Feldes.
34	CRC-End	rezessives CRC-Begrenzungsbit (cyclic redundancy code)
35	ACK-Slot	Acknowledge Slot für die stationsneutrale posititve Empfangsbestätigung des CAN-Protokolls ist von zwei rezessiven Bits eingerahmt. Der Sender legt ein rezessives Bit an und erwartet die Empfangsbestätigung in Form eines domianten Pegels durch mindestens einen Empfänger.
36	ACK-End	Acknowledge Delimiter, rezessives Begrenzungsbit
37-43	EOF-Field	Jeder Daten- und Datenanforderungsframe wird durch 7 rezessive End of Frame Bits beendet. Mit dem ACK-End Bit beenden 8 rezessive Bits die Nachricht.
44-46	Intermission	ACK-End + EOF-Field + Intermission Field ergibt mindestens 11 rezessive Bits InterFrame Space.

CAN Error Detection

Das CAN-Protokoll hat 5 Fehlererkennungsmechanismen implementiert.

Die 2 Fehlererkennungsmechanismen auf Bit-Level:

- Bit Stuffing: Netzknoten erkennt die Verletzung der Bit Stuffing Codierungsregeln.
- Bit Monitoring: Vergleich von Sende- und Empfangspegel durch den Sender.

Die 3 Fehlererkennungsmechanismen auf Frame-Level:

- Cyclic Redundancy Checks (CRC): Berechnung einer 15 Bit Prüfsequenz über SOF bis zum letzten Bit des Datenfeldes.
- Frame Checks: Netzknoten erkennt Formatfehler des Frames, wenn Begrenzungsbits fehlerhaft gesetzt.
- Acknowledgement Checks: Sender erwartet im ACK-Slot die Empfangsbestätigung seiner Nachricht.

CAN Error Confinement

Zur Fehlerbegrenzung sind in jedem Netzwerkknoten Zähler für die erkannten Sende- und Empfangsfehler implemetiert. In Abhängigkeit der aktuellen Zählerstände befinden sich die Netzknoten in einem von drei Zuständen:

- Active Error: normale Kommunikation der Station, senden des Aktive Error Flags im Fehlerfall
- Passive Error: verzögerte Kommunikation der Station, senden des Passive Error Flags im Fehlerfall
- Bus off: Station ist durch die Bustreiber vom Bus abgeschaltet