

Controller Area Network

Dominik Eisele

Werner-Siemens-Schule

30. Juni 2016

Inhalt

Allgemeine Informationen

Protokoll

Anwendungen

Quellen

Geschichte

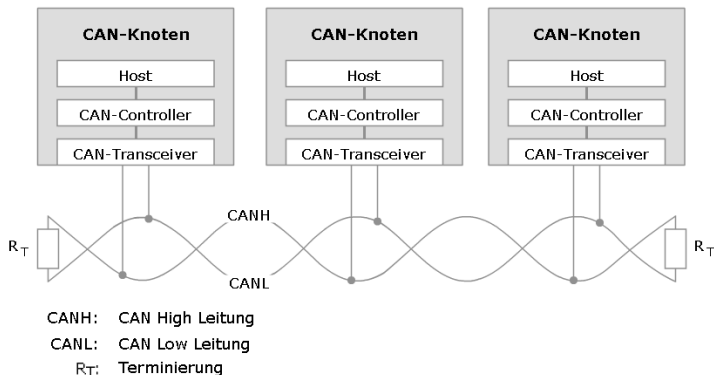
- 1983 von Bosch als seriellcs Feldbussystem entwickelt
- Ziel war die Reduzierung der Kabelbaumlänge in Fahrzeugen
- Zertifiziert nach ISO 11898-2 und ISO 11898-3 (High- und Low-Speed CAN)
- CAN besitzt eine sehr sichere Datenübertragung, welche Echtzeitanforderungen gerecht wird

Aufbau



CAN-Kommunikation

CAN-Netzwerk

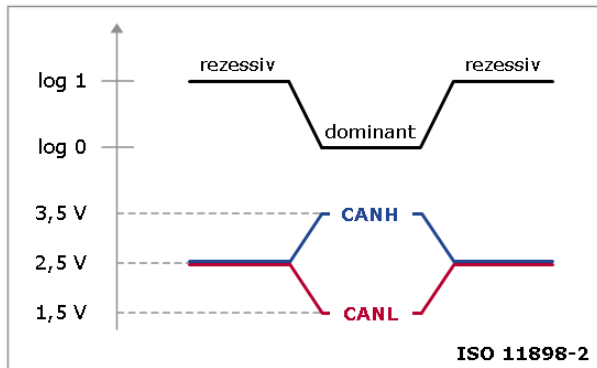


CAN-Buspegel



CAN-Kommunikation

CAN-Highspeed-Buspegel

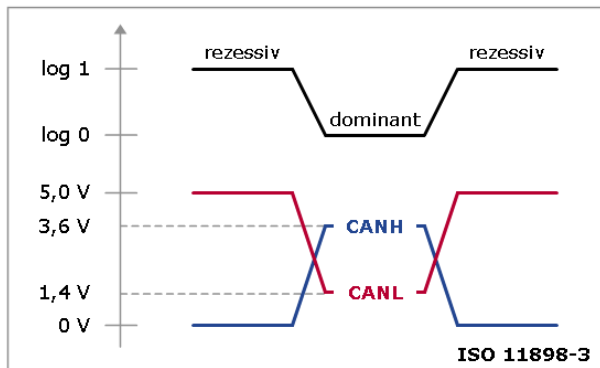


CAN-Buspegel



CAN-Kommunikation

CAN-Lowspeed-Buspegel



Arbitrierung

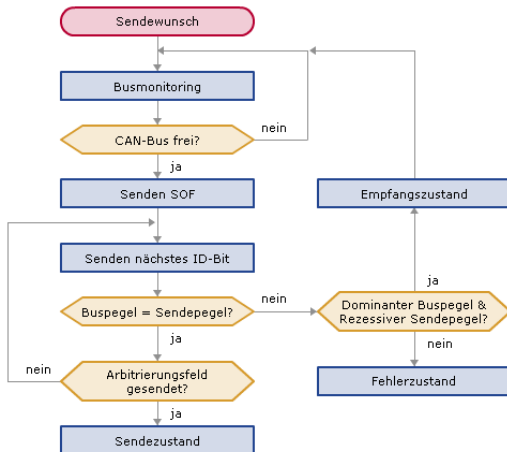
- CAN ist ein Multimasterbus, d.h. Teilnehmer müssen selbst entscheiden wann sie senden
- zum Einsatz kommt daher das Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) Verfahren
- wenn zwei Teilnehmer gleichzeitig senden kommt die bitweise Arbitrierung zum Einsatz
- bei der Arbitrierung werden die Identifier gleichzeitig gesendet und der Buspegel mit dem Sendepiegel verglichen
- sendet ein Teilnehmer ein dominantes und ein anderer ein rezessives Bit wird der Buspegel dominant (logische 0)

Arbitrierung

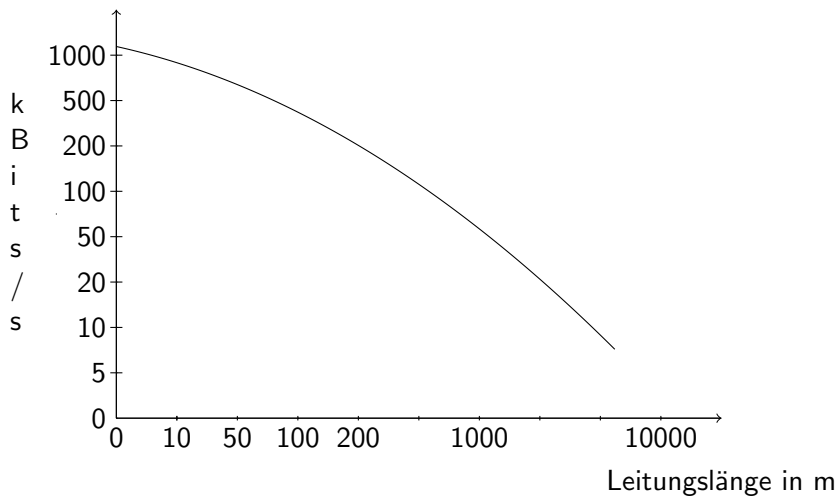


Buszugriff im CAN-Netzwerk

Buszugriffsprozedur



Leitungslänge

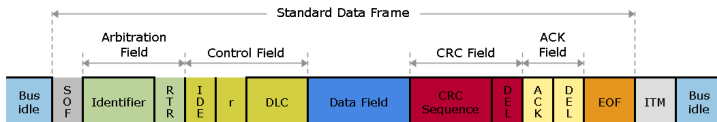


CAN-Frame



CAN-Framing

Standard Data Frame

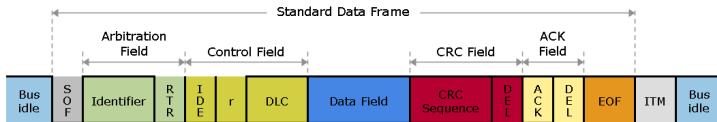


CAN-Frame



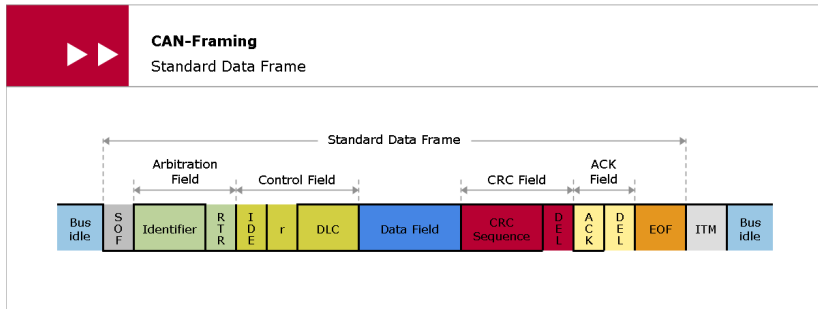
CAN-Framing

Standard Data Frame



- Start of Frame Bit ist ein dominantes Bit dass den Beginn einer Nachricht markiert
- wird zur zeitlichen Synchronisation der Knoten genutzt

CAN-Frame



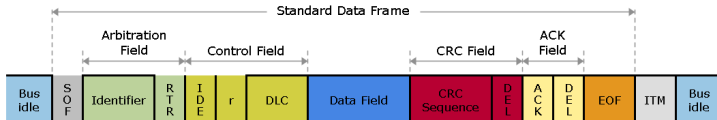
- Identifier ist 11 bit lang
- Extended Identifier hat 18 Zusatzbits
- Identifier markieren die Priorität der Nachricht

CAN-Frame



CAN-Framing

Standard Data Frame



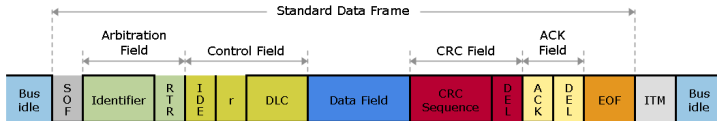
- Remote Transmission Request Bit
- besitzt kein Data Field, sondern fordert Nachricht mit gleicher ID an wenn RTR rezessiv
- problematisch, da es unterschiedliche Implementierungen gibt

CAN-Frame



CAN-Framing

Standard Data Frame



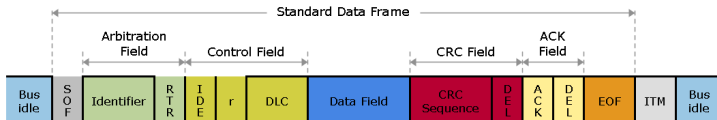
- Identifier Extension Bit ist normal dominant
- wenn IDE rezessiv folgt erweiterter Identifier (18 Bit)
- Protokolleffektivität wird verschlechtert wenn IDE verwendet wird

CAN-Frame



CAN-Framing

Standard Data Frame



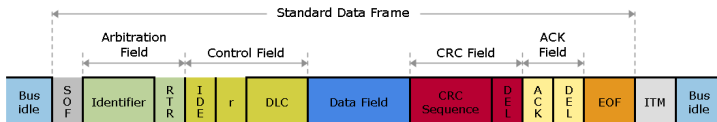
- Data Length Code gibt die Länge des Data Field an
- DLC sind 4 Bit

CAN-Frame



CAN-Framing

Standard Data Frame



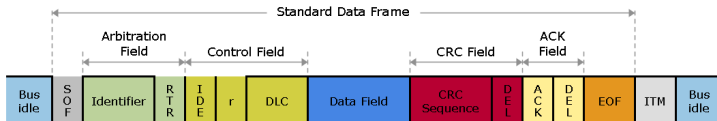
- Data Field beinhaltet die Daten
- Länge wird in DLC von 0 bis 8 Byte festgesetzt

CAN-Frame



CAN-Framing

Standard Data Frame



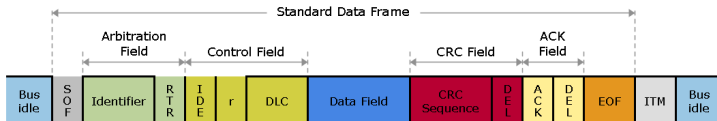
- Cyclic Redundancy Check is eine 15 Bit lange Prüfsequenz
- Mit der, in der Prüfsequenz enthaltenen redundanten Information kann ein Empfänger prüfen, ob die Nachricht verfälscht wurde.
- wird von rezisivem Delimiter-(DEL) Bit begrenzt

CAN-Frame



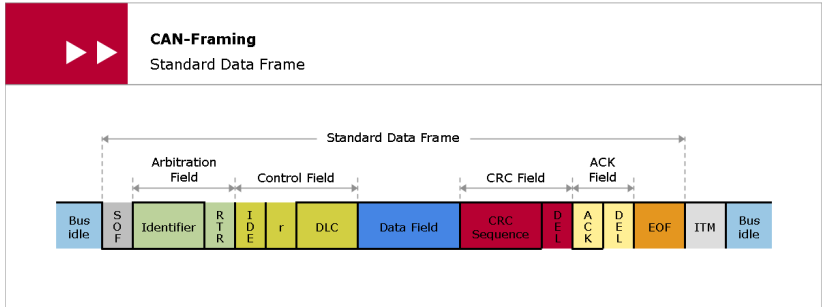
CAN-Framing

Standard Data Frame



- Acknowledge Field wird rezessiv gesendet, und bei korrektem Empfang von einem Empfänger dominant überschrieben
- wenn ACK rezessiv ist, wird die Nachricht erneut übertragen
- wird bei einem Empfänger ein Fehler erkannt, das ACK Bit ist aber gesetzt, so sendet er im Anschluss ein Error-Flag um die Nachricht erneut anzufordern, und die systemweite Datenkonsistenz zu wahren

CAN-Frame



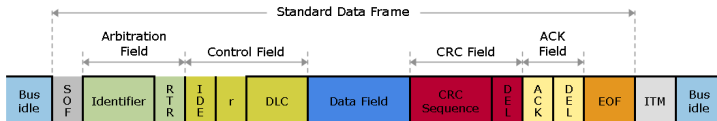
- Delimiter begrenzt ACK-Feld rezissiv
- wird bei lokalem Fehler eines Empfängers mit Error-Flag überschrieben, sodass die Nachricht erneut gesendet wird
- dient der Unterscheidung zwischen Error-Flag und ACK-Feld Ende

CAN-Frame



CAN-Framing

Standard Data Frame



- End of Frame Sequenz wird am Ende jeder Nachricht gesendet und besteht aus 8 rezessiven Bits
- lange Sequenz nötig um Empfängern die Möglichkeit zu geben Fehler zu erkennen und zu Melden

Bit Stuffing

- nach 5 gleichen Bits muss ein inverses Bit eingefügt werden
- dient der Nachsynchronisierung der Busteilnehmer, da sie aufgrund ungenauen Taktzeiten der ICs auseinanderdriften können
- dient ausserdem der Unterscheidung von Errorflags
- EOF ist vom Bitstuffing ausgenommen
- die Stopf-Bits werden vom Transmitter herausgefiltert

Fehlererkennung

- Bit Monitoring
- Bit Stuffing
- Frame Check
- Acknowledgement Check
- Cyclic Redundancy Check

Fehlermeldung

- jeder Empfänger versucht einen Fehler zu erkennen und diesen umgehend zu dementieren
- Fehlermeldungen werden sofort nach erkennen des Fehlers gesendet
- ausgenommen hiervon ist das CRC-Feld um die ACK Funktion aufrecht zu erhalten
- Fehlermeldungen bestehen aus 6 gleichen Bits (rezessiv oder dominant)
- das Error-Flag zerstört den Bus Traffic
- wenn ein Fehler erkannt wird werden bei allen Empfängern die Nachrichten gelöscht, um eine systemweite Datenkonsistenz aufrecht zu erhalten

Bus Zustände

- jeder Knoten hat zwei Error Counter (Transmit Error Counter und Error Counter)
- sendet ein Knoten ein Error Flag erhöht sich der Counter um 8, empfängt er ein Flag erhöht er sich um 1, bei korrektem Empfang wird er um 1 dekrementiert
- ist der Counter kleiner 127 sendet er als Error Flag 6 dominante Bits, die den Bus überschreiben und somit alle Nachrichten überschreiben
- ist der Counter größer 127 werden 6 rezessive Bits gesendet, sodass der Bus nicht durchgehend gestört wird
- ist der Counter bei 255 angelangt darf der Teilnehmer nicht mehr am Bus teilnehmen

Anwendungsbeispiele

- Automobilindustrie
- Automatisierungstechnik
- Aufzugsanlagen
- Medizintechnik
- Flugzeugtechnik
- Raumfahrttechnik
- Schienenfahrzeuge
- Schiffbau

Quellen

- Konrad Etschberger - CAN Controller-Area-Network
Grundlegen, Protokolle, Bausteine, Anwedungen
- Praxis Profiline - Controller-Area-Network CAN in
Automation (CiA)
- elearning.vector.com (30.06.2016)
- kvaser.com (30.06.16)
- wikipedia.com (30.06.16)