CAN Bus over Ethernet

R. Löffler, C. Proske, M. Sharkhawy

21. Juni 2014

Inhaltsverzeichnis

1 CAN Bus **2**
- 1.1 CAN Geschichte . . . 2
- 1.2 Busankoplung/Vernetzung von CAN-Knoten . . . 2
- 1.3 CAN-Bus Pegel . . . 2
- 1.4 Bit Stuffing . . . 2
- 1.5 CAN Frames . . . 2
- 1.6 Zeichenkodierung . . . 5
- 1.7 Bit-Timing . . . 5
- 1.8 Synchronisatzion . . . 5
- 1.9 Buszugriffsverfahren . . . 5
- 1.10 Fehlermanagement . . . 5
- 1.11 Sicherungsmechanismen/Fehlererkennung . . . 5

2 CAN Bus over Ethernet **5**
- 2.1 Probleme . . . 5
- 2.2 Bestehende Lösungen . . . 5

Abbildungen **6**

Literatur **7**

1 CAN Bus

1.1 CAN Geschichte

1.2 Busankoplung/Vernetzung von CAN-Knoten

1.3 CAN-Bus Pegel

Beim CAN-Bus erfolgt die Übertragung der Bits über 2 Pegel:

- Dominant
- Rezessiv

Wenn beide Pegel gleichzeitig gesendet werden, so überschreibt immer der dominante Pegel den rezessiven Pegel.

1.4 Bit Stuffing

Bit-Stuffing beim CAN-Bus bedeutet, dass nach 5 Bits mit gleichem Pegel ein »Stuff Bit« mit inversem Pegel eingefügt wird. Die Empfänger filtern diese Stuff-Bits, dann nach dem gleichen Schema wieder heraus.
Bit stuffing wird einerseits eingesetzt, weil Bitfolgen mit mehr als 5 Bits mit gleichem Pegel für Steuerungszwecke eingesetzt werden, und andererseits auch wegen der NRZ-Kodierung. Das heißt dass der Pegel bei zB 4 rezessiven Bits gleich bleibt. Wenn dann beispielsweise 10 gleiche Bits übertragen werden, so kann der Empfänger möglicherweise nicht mehr unterscheiden, ob jetzt 10 oder 11 Bits übertragen wurden.

1.5 CAN Frames

Beim CAN-Bus gibt es 4 unterschiedliche Arten von Frames:

- Daten-Frame
- Remote-Frame
- Error-Frame
- Overload-Frame

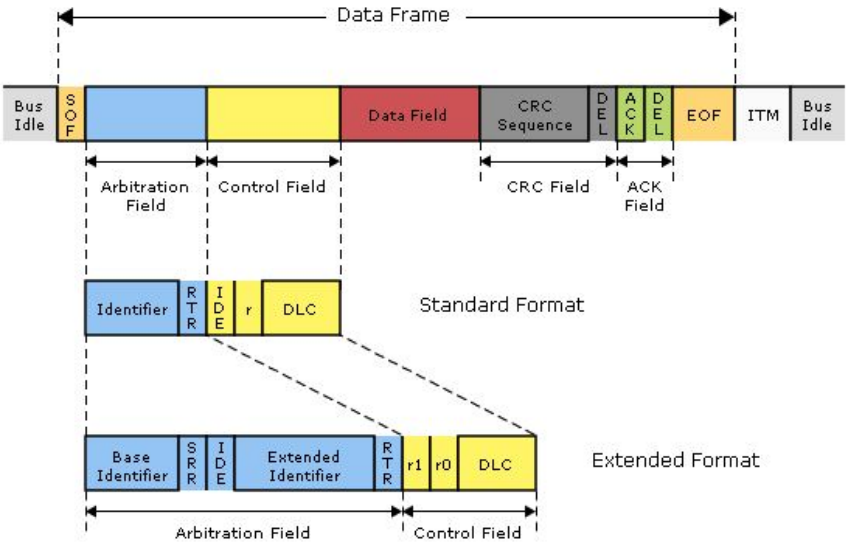

Abbildung 1: Aufbau eines Daten-Frames (2)

Daten Frame

Die Daten-Frames dienen dem Transport von Daten und können bis zu 8 Byte an Nutzdaten enthalten. Bei den Daten-Frames kann zwischen Standard Format und Extended Format unterschieden werden, die sich hauptsächlich in der Länge des Identifiers unterscheiden. Der Aufbau eines Daten Frames kann aus Abbildung 1 entnommen werden.

Gestarted wird ein Frame durch das **SOF** (start-of-frame) bestehend aus einem dominanten Bit. Darauf folgt gleich das Arbitrationsfeld und dem Kontrollfeld. Abhängig vom Format bestehen diese aus **Basis Identifier** (11 Bits), **Extended Identifier** (18 Bit), einem **RTR**-Bit (remote transmission request; bei Remote-Frames rezessiv) sowie einem **SRR**-Bit (substitude-remote-request; rezessiv). Ist das **IDE**-Bit (identifier extension) rezessiv, so handelt es sich um das Extended Format, das somit immer Nachrang über dem Standard Format hat.
Im Kontrollfeld befindet sich neben 1-2 reservierten (aktuell nicht verwendeten) Bits der aus 4 Bits bestehende **DLC** (data length code), der die Länge des nachfolgenden **Datenfeldes** angibt.
Wie bereits erwähnt können mit den Daten-Frames bis zu 8 Byte, also 64 Bit, übertragen werden. dies kann jedoch nur in einheiten von 8 Bits geschehen. Anschließend an das Datenfeld ist das CRC-Feld (cyclic redundancy check; Prüfsummenfeld) bestehend aus 16 Bit (15 Bit + 1 rezessives Delimiter-Bit). Des weiteren enthält ein Daten-Frame ein **ACK**-Feld (Acknowledge). Dieses wird verwendet, um den Empfang eines korrekten Frames zu bestätigen. Der Sender sendet dafür ein rezessives Bit. Jeder Empfänger der keinen Fehler feststellen konnte, setzt einen dominanten Pegel und überschreibt somit den rezessiven des Senders. Im Falle einer negativen Quittierung (rezessiver Pegel) muss der Knoten, der den Fehler erkannt hat, nach dem ACK-Delimiter einen Error-Frame aussenden.
Das Ende des Frames wird mit 7 rezessiven Bits dem **EOF** (end of frame) angezeigt.

Anschließend an den Frame muss ein ITM-Package (intermission oder inter-frame-space) bestehend aus mindestens 3 rezessiven Bits gesendet werden.

Remote Frame

Mit Hilfe der Remote-Frames können Daten-Frames von anderen Teilnehmern angefordert werden. Der Frame unterscheidet sich vom Daten-Frame durch ein rezessives Bit im »RTR«-Slot, wodurch Remote-Frames im Falle einer Kollision immer Nachrang gegenüber den Daten-Frames haben, und durch ein Fehlen des Datenfeldes.

Error Frame

Erkennt ein CAN-Knoten einen Fehler, so sendet er einen Error-Frame an alle anderen CAN-Knoten im Netzwerk. Bei diesen Frames wird das Bit-Stuffing bewusst ignoriert. Der Error-Frame besteht aus 2 Feldern, den Error-Flags und dem Error-Delimiter (8 rezessive Bits). Die Error-Flags sind abhängig vom Modus in dem sich ein CAN-knoten befindet. Ist der Knoten im Fehler-Status *»error active«* so setzt er die Error-Flags auf 6 dominante Bits. Befindet er sich hingegen im Fehler-Status *»error passive«* so sendet er 6 rezessive Bits.

Overload Frame

Overload-Frames werden als Zwangspause zwischen Daten- und Remote-Frames genutzt. Dabei hat ein Overload-Frame das gleiche Format wie ein Active-Error-Frame. Wie in Abbildung 2 ersichtlich ist, besteht der Overload-Frame ebenfalls aus 2 Feldern: dem Overload-Flag (6 dominante Bits) und dem Overload-Delimiter (8 rezessive Bits). Ein Overload-Frame kann jedoch nur während eines Interframespaces gesendet werden. Dies ermöglicht die Unterscheidung von den Error-Frames, die während der Übertragung einer Nachricht gesendet werden, sobald eben ein Fehler erkannt wurde.

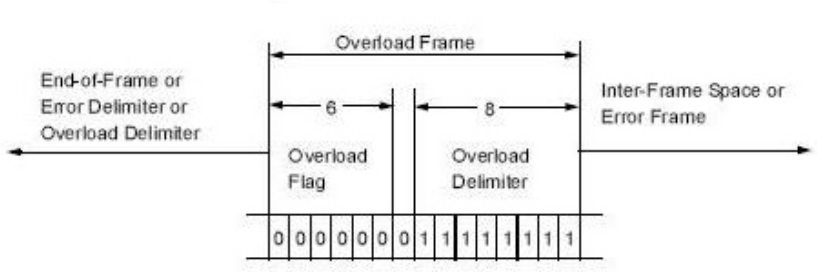

Abbildung 2: Aufbau eines Overload-Frames (1)

1.6 Zeichenkodierung

1.7 Bit-Timing

1.8 Synchronisatzion

1.9 Buszugriffsverfahren

1.10 Fehlermanagement

1.11 Sicherungsmechanismen/Fehlererkennung

2 CAN Bus over Ethernet

2.1 Probleme

2.2 Bestehende Lösungen

Abbildungsverzeichnis

1 Aufbau eines Daten-Frames (2) 3
2 Aufbau eines Overload-Frames (1) 4

Literatur

[1] Can bus message frames - overload frame,interframe space.

[2] M. Guardigli. Hacking your car.