

# Controller Area Network

Dominik Eisele

Werner-Siemens-Schule

29. Juni 2016

# Inhalt

Allgemeine Informationen

Protokoll

Anwendungen

Quellen

# Geschichte

- 1983 von Bosch als seriellles Feldbussystem entwickelt
- Ziel war die Reduzierung der Kabelbaumlänge in Fahrzeugen

# 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)

# 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

# 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

# Addition von Dualzahlen

Rechenbeispiel für eine Addition mit Dual-Zahlen.

$$\begin{array}{r} 111010010 \\ + 001110100 \\ \hline 1001000110 \end{array}$$

# Subtraktion von Dualzahlen

Rechenbeispiel für eine Subtraktion mit Dual-Zahlen.

$$\begin{array}{r} 111010010 \\ - 001110100 \\ \hline 101011110 \end{array}$$



# Subtraktion mit Hilfe des Zweierkomplements

Da, in der Digitaltechnik, für die Subtraktion von Dualzahlen keine logische Verknüpfung existiert, ist man gezwungen eine Subtraktion in eine Addition umwandeln.

$$2 - 6 = (-4)$$

# Subtraktion mit Hilfe des Zweierkomplements

Da, in der Digitaltechnik, für die Subtraktion von Dualzahlen keine logische Verknüpfung existiert, ist man gezwungen eine Subtraktion in eine Addition umwandeln.

$$2 - 6 = (-4)$$

$$2 + (-6) = (-4)$$

# Subtraktion mit Hilfe des Zweierkomplements

Da, in der Digitaltechnik, für die Subtraktion von Dualzahlen keine logische Verknüpfung existiert, ist man gezwungen eine Subtraktion in eine Addition umwandeln.

$$2 - 6 = (-4)$$

$$2 + (-6) = (-4)$$

# Subtraktion mit Hilfe des Zweierkomplements

$$\underline{2 - 6 = ?}$$

1. Schritt: In eine Dualzahl wandeln:

$$2 - 6 \Rightarrow 10 - 110$$

# Subtraktion mit Hilfe des Zweierkomplements

$$\underline{2 - 6 = ?}$$

1. Schritt: In eine Dualzahl wandeln:

$$2 - 6 \Rightarrow 10 - 110$$

2. Schritt: Stellen auffüllen:

$$0010 - 0110 = ?$$

## Subtraktion mit Hilfe des Zweierkomplements

$$\underline{2 - 6 = ?}$$

1. Schritt: In eine Dualzahl wandeln:

$$2 - 6 \Rightarrow 10 - 110$$

2. Schritt: Stellen auffüllen:

$$0010 - 0110 = ?$$

3. Schritt: Bits negieren:

$$0110 \Rightarrow 1001$$

## Subtraktion mit Hilfe des Zweierkomplements

$$\underline{2 - 6 = ?}$$

1. Schritt: In eine Dualzahl wandeln:

$$2 - 6 \Rightarrow 10 - 110$$

2. Schritt: Stellen auffüllen:

$$0010 - 0110 = ?$$

3. Schritt: Bits negieren:

$$0110 \Rightarrow 1001$$

4. Schritt: Hinzuaddieren von 1:

$$1001 + 0001 = 1010$$

## Subtraktion mit Hilfe des Zweierkomplements

$$\underline{2 - 6 = ?}$$

1. Schritt: In eine Dualzahl wandeln:

$$2 - 6 \Rightarrow 10 - 110$$

2. Schritt: Stellen auffüllen:

$$0010 - 0110 = ?$$

3. Schritt: Bits negieren:

$$0110 \Rightarrow 1001$$

4. Schritt: Hinzuaddieren von 1:

$$1001 + 0001 = 1010$$

5. Schritt: Minuend und Zweierkomplement addieren:

$$0010 + 1010 = 1100$$



## Subtraktion mit Hilfe des Zweierkomplements

$$\underline{2 - 6 = ?}$$

1. Schritt: In eine Dualzahl wandeln:

$$2 - 6 \Rightarrow 10 - 110$$

2. Schritt: Stellen auffüllen:

$$0010 - 0110 = ?$$

3. Schritt: Bits negieren:

$$0110 \Rightarrow 1001$$

4. Schritt: Hinzuaddieren von 1:

$$1001 + 0001 = 1010$$

5. Schritt: Minuend und Zweierkomplement addieren:

$$0010 + 1010 = 1100$$

6. Schritt: Ergebnis negieren:

$$100 \Rightarrow 011$$

## Subtraktion mit Hilfe des Zweierkomplements

$$\underline{2 - 6 = ?}$$

1. Schritt: In eine Dualzahl wandeln:

$$2 - 6 \Rightarrow 10 - 110$$

2. Schritt: Stellen auffüllen:

$$0010 - 0110 = ?$$

3. Schritt: Bits negieren:

$$0110 \Rightarrow 1001$$

4. Schritt: Hinzuaddieren von 1:

$$1001 + 0001 = 1010$$

5. Schritt: Minuend und Zweierkomplement addieren:

$$0010 + 1010 = 1100$$

6. Schritt: Ergebnis negieren:

$$100 \Rightarrow 011$$

# Subtraktion mit Hilfe des Zweierkomplements

7. Schritt: Hinzuaddieren von 1:

$$011 + 001 = 100$$

8. Schritt: In eine Dezimalzahl wandeln:

$100 \Rightarrow 4$  ; da das höchstwertigste Bit 1 ist:

Endergebnis =  $-4$

# Subtraktion mit Hilfe des Zweierkomplements

7. Schritt: Hinzuaddieren von 1:

$$011 + 001 = 100$$

8. Schritt: In eine Dezimalzahl wandeln:

$100 \Rightarrow 4$  ; da das höchstwertigste Bit 1 ist:

Endergebnis =  $-4$

9. Schritt: Ergebnis:

$$\underline{2 - 6 = (-4)}$$

# Subtraktion mit Hilfe des Zweierkomplements

7. Schritt: Hinzuaddieren von 1:

$$011 + 001 = 100$$

8. Schritt: In eine Dezimalzahl wandeln:

$100 \Rightarrow 4$  ; da das höchstwertigste Bit 1 ist:

Endergebnis =  $-4$

9. Schritt: Ergebnis:

$$\underline{2 - 6 = (-4)}$$

# Multiplikation von Dualzahlen

- Es werden Produkte mit den einzelnen Stellen des Multiplikators gebildet, sie werden anschließend Stellenrichtig addiert.
- Die Stellen des Multiplikators können nur Zahlenwerte zwischen Null und Eins annehmen

# Multiplikation von Dualzahlen

- Es werden Produkte mit den einzelnen Stellen des Multiplikators gebildet, sie werden anschließend Stellenrichtig addiert.
- Die Stellen des Multiplikators können nur Zahlenwerte zwischen Null und Eins annehmen
- → Dies kann mit einer einfachen UND-Verknüpfung gelöst werden.

# Multiplikation von Dualzahlen

- Es werden Produkte mit den einzelnen Stellen des Multiplikators gebildet, sie werden anschließend Stellenrichtig addiert.
- Die Stellen des Multiplikators können nur Zahlenwerte zwischen Null und Eins annehmen
- → Dies kann mit einer einfachen UND-Verknüpfung gelöst werden.



# Multiplikation von Dualzahlen

Rechenbeispiel für eine Multiplikation mit Dual-Zahlen.

$$\begin{array}{r} 1011 \times 1010 \\ \hline 1011000 \\ 000000 \\ 10110 \\ 0000 \\ + \\ \hline 1101110 \end{array}$$

## Halbaddierer

Ein Halbaddierer besitzt zwei Ein-, und zwei Ausgänge. An die Eingänge  $x$  und  $y$  werden jeweils die Ziffern angelegt die man addieren möchte. An dem ersten Ausgang liegt die Summe  $s$  der Addition an, am zweiten Ausgang der Übertrag  $c$ .

$x$	$y$	Übertrag $c$	Summe $s$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

# Schaltbild Halbaddierer

In Schaltungen wird der Halbaddierer aus zwei Bauteilen zusammengesetzt, ein Exklusiv-ODER (XOR) und ein UND (AND).

# Volladdierer

- Der Volladdierer besteht aus zwei Halbaddierern und einem ODER.
- Der Volladdierer hat einen zusätzlichen Eingang ( $c_{in}$ ), man kann den Übertrag aus einer vorhergegangenen Addition mit einbeziehen.

# Volladdierer

- Der Volladdierer besteht aus zwei Halbaddierern und einem ODER.
- Der Volladdierer hat einen zusätzlichen Eingang ( $c_{in}$ ), man kann den Übertrag aus einer vorhergegangenen Addition mit einbeziehen.
- Man kann mehrere Volladdierer kaskadieren um größere Zahlen miteinander zu addieren.

x	y	$c_{in}$	$c_{out}$	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

# Volladdierer

- Der Volladdierer besteht aus zwei Halbaddierern und einem ODER.
- Der Volladdierer hat einen zusätzlichen Eingang ( $c_{in}$ ), man kann den Übertrag aus einer vorhergegangenen Addition mit einbeziehen.
- Man kann mehrere Volladdierer kaskadieren um größere Zahlen miteinander zu addieren.

x	y	$c_{in}$	$c_{out}$	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

## Schaltbild Volladdierer

In Schaltungen wird der Halbaddierer aus zwei Bauteilen zusammengesetzt, ein Exklusiv-ODER (XOR) und ein UND (AND).

## Subtrahierer

Beim Subtrahierer wird der Volladdierer durch einen Steuereingang erweitert. An diesem legt man für eine Addition eine 0, und für eine Subtraktion eine 1 an. Durch diesen Steuereingang wird das Zweierkomplement gebildet.



# Multiplizierer

- Der Multiplizierer kann zwei acht Bit Zahlen miteinander multiplizieren.
- Dazu wird die Zahl an ein Schieberegister angelegt, sodass sie bei jedem Takt um eine Stelle verschoben werden kann.

# Multiplizierer

- Der Multiplizierer kann zwei acht Bit Zahlen miteinander multiplizieren.
- Dazu wird die Zahl an ein Schieberegister angelegt, sodass sie bei jedem Takt um eine Stelle verschoben werden kann.
- Die höchstwertigste Ziffer des Multiplikators wird abgegriffen, und mit dem Multiplikant multipliziert.

# Multiplizierer

- Der Multiplizierer kann zwei acht Bit Zahlen miteinander multiplizieren.
- Dazu wird die Zahl an ein Schieberegister angelegt, sodass sie bei jedem Takt um eine Stelle verschoben werden kann.
- Die höchstwertigste Ziffer des Multiplikators wird abgegriffen, und mit dem Multiplikant multipliziert.
- Um die Produkte Stellenrichtig zu addieren werden die Zwischenergebnisse ebenfalls verschoben.

# Multiplizierer

- Der Multiplizierer kann zwei acht Bit Zahlen miteinander multiplizieren.
- Dazu wird die Zahl an ein Schieberegister angelegt, sodass sie bei jedem Takt um eine Stelle verschoben werden kann.
- Die höchstwertigste Ziffer des Multiplikators wird abgegriffen, und mit dem Multiplikant multipliziert.
- Um die Produkte Stellenrichtig zu addieren werden die Zwischenergebnisse ebenfalls verschoben.
- Das Produkt erhält man, nach acht Takten Rechenzeit.

# Multiplizierer

- Der Multiplizierer kann zwei acht Bit Zahlen miteinander multiplizieren.
- Dazu wird die Zahl an ein Schieberegister angelegt, sodass sie bei jedem Takt um eine Stelle verschoben werden kann.
- Die höchstwertigste Ziffer des Multiplikators wird abgegriffen, und mit dem Multiplikant multipliziert.
- Um die Produkte Stellenrichtig zu addieren werden die Zwischenergebnisse ebenfalls verschoben.
- Das Produkt erhält man, nach acht Takten Rechenzeit.

## BCD-Eingabe

- Die Eingabe erfolgt über ein ASCII-Feld.
- Bei einer Eingabe von einer Ziffer im Bereich 0 bis 9, liegt die Ziffer BCD codiert an den Ausgänge  $D_0$  bis  $D_3$  an

## BCD-Eingabe

- Die Eingabe erfolgt über ein ASCII-Feld.
- Bei einer Eingabe von einer Ziffer im Bereich 0 bis 9, liegt die Ziffer BCD codiert an den Ausgänge  $D_0$  bis  $D_3$  an
- Nach jedem Tastendruck wird die Ziffer in ein Register übernommen, man kann an ihren Ausgängen die Zahl von 0 bis 99 abgreifen.

## BCD-Eingabe

- Die Eingabe erfolgt über ein ASCII-Feld.
- Bei einer Eingabe von einer Ziffer im Bereich 0 bis 9, liegt die Ziffer BCD codiert an den Ausgänge  $D_0$  bis  $D_3$  an
- Nach jedem Tastendruck wird die Ziffer in ein Register übernommen, man kann an ihren Ausgängen die Zahl von 0 bis 99 abgreifen.



# Umwandlung

- Für die Wandlung werden zwei Zähler und ein Wandler benötigt.
- Der erste Zähler ist ein BCD-Zähler, der zweite ein Binär-Zähler.

# Umwandlung

- Für die Wandlung werden zwei Zähler und ein Wandler benötigt.
- Der erste Zähler ist ein BCD-Zähler, der zweite ein Binär-Zähler.
- Beide Zähler haben den selben Clock.

# Umwandlung

- Für die Wandlung werden zwei Zähler und ein Wandler benötigt.
- Der erste Zähler ist ein BCD-Zähler, der zweite ein Binär-Zähler.
- Beide Zähler haben den selben Clock.
- Der Vergleicher vergleicht das eingegebene Signal mit dem BCD-Zähler.

# Umwandlung

- Für die Wandlung werden zwei Zähler und ein Wandler benötigt.
- Der erste Zähler ist ein BCD-Zähler, der zweite ein Binär-Zähler.
- Beide Zähler haben den selben Clock.
- Der Vergleicher vergleicht das eingegebene Signal mit dem BCD-Zähler.
- Sind beide Signale identisch wird der Clock unterbrochen.

# Umwandlung

- Für die Wandlung werden zwei Zähler und ein Wandler benötigt.
- Der erste Zähler ist ein BCD-Zähler, der zweite ein Binär-Zähler.
- Beide Zähler haben den selben Clock.
- Der Vergleicher vergleicht das eingegebene Signal mit dem BCD-Zähler.
- Sind beide Signale identisch wird der Clock unterbrochen.
- Man kann das Binär-Ergebnis am Ausgang des Binär-Zählers ablesen.

# Umwandlung

- Für die Wandlung werden zwei Zähler und ein Wandler benötigt.
- Der erste Zähler ist ein BCD-Zähler, der zweite ein Binär-Zähler.
- Beide Zähler haben den selben Clock.
- Der Vergleicher vergleicht das eingegebene Signal mit dem BCD-Zähler.
- Sind beide Signale identisch wird der Clock unterbrochen.
- Man kann das Binär-Ergebnis am Ausgang des Binär-Zählers ablesen.

## 7-Segment-Anzeige

- Das Endergebnis wird Ziffer für Ziffer auf ein 7-Segment-Decoder gegeben.
- Die Ausgänge des 7-Segment-Decoders werden auf die Eingänge einer 7-Segment-Anzeige gegeben.

## 7-Segment-Anzeige

- Das Endergebnis wird Ziffer für Ziffer auf ein 7-Segment-Decoder gegeben.
- Die Ausgänge des 7-Segment-Decoders werden auf die Eingänge einer 7-Segment-Anzeige gegeben.



- Junior Computer - Der einfache Einstieg in die faszinierende Computertechnik (1980)
- [www.elektronik-kompodium.de/sites/dig/1708061.htm](http://www.elektronik-kompodium.de/sites/dig/1708061.htm) (11.04.2015)
- [de.wikipedia.org/wiki/Volladdierer](http://de.wikipedia.org/wiki/Volladdierer) (16.05.2015)