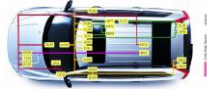


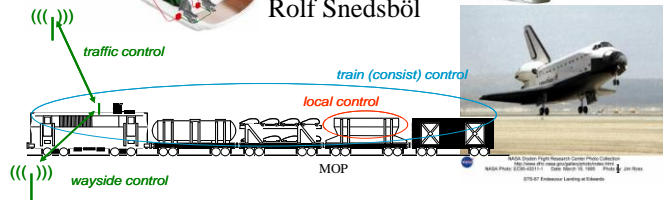


# CAN

ett kommunikationsprotokoll



Rolf Snedsböl



## CAN Controller Area Network

- Meddelanden, allmänt
- Bussåtkomst
- Synkronisering
- Olika ramar
- Felhantering
- Identifierare

1984, 1991 CAN C-Spec, Bosch och Intel

- Kommunikationsstandard för bilindustrin
- Standard CAN, senare Extended CAN

Implementerat i hårdvara

- INTEL, MOTOROLA, PHILIPS, Siemens, NCS, SGS, etc,etc

Protokoll för distribuerad styrning i händelsestyrda system med små mängder data (1-8 bytes) och ett datanät med fysisk liten utsträckning.

MOP

2

## CAN-protokollets egenskaper

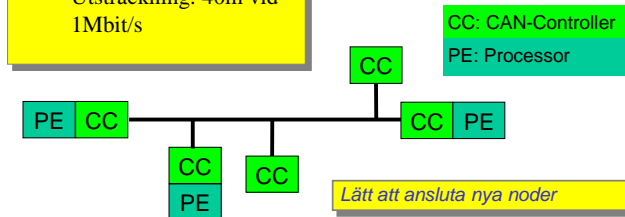
- Flexibelt
- Multimaster protokoll
- Multicast protokoll
- Hög överföringshastighet (1 Mbit/s)
- Kort svarstid (200  $\mu$ s)
- Automatisk omsändning av störda överföringar
- ”Atomic Broadcast”
- Stöd för synkroniserad exekvering i olika noder
- Avlastar processorn/användaren med meddelandeöverföringen

MOP

3

## Ett typiskt CAN nät

- Busstopologi
- Media: twisted pair, koaxial, fiber
- Utsträckning: 40m vid 1Mbit/s

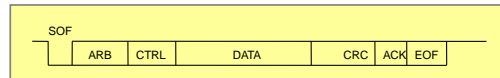


MOP

4

# CAN- meddelandet

- Objekt (meddelanden)
  - Ex: broms, ljus, fönsterhiss
  - Skicka/efterfråga ett objekt
- Unik prioritet knuten till identifieraren
  - Specificeras innan "run-time"
  - Prioriteten kan tolkas som en "adress"



- ARB** Arbitrering (identifierare)
- CTRL** Kontrollinformation
- DATA** 0-8 bytes
- CRC** Kontrollsumma
- ACK** Kvitto (Inbyggd handsakning)
- EOF** Slutmarkering

MOP

5

# Bussåtkomst

## CSMA/CD (Ethernet)

- Carrier Sense: Lyssna Först
- Multiple Access: Vem som helst kan skicka ett meddelande när bussen är ledig.
- Collision Detect: Buss-krock upptäcks.

Algoritm: Börja om vid fel

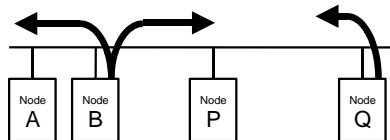
MOP

6

# Bussåtkomst forts

## Algoritm (CSMA/CD)

- Lyssna på bussen;
- Om bussen är ledig starta sändning av ett meddelande



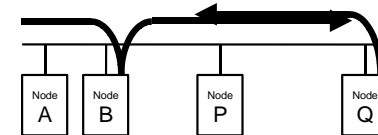
MOP

7

# Bussåtkomst forts

## Algoritm (CSMA/CD)

- Lyssna på bussen och kontrollera om busskrock uppstår.



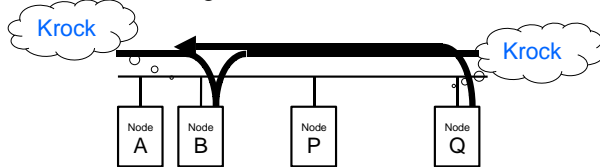
MOP

8

## Bussåtkomst forts

### Algoritm (CSMA/CD)

- Om busskrock uppstått, avsluta då sändningen och försök igen senare.



MOP

9

## Bussåtkomst forts

### •Algoritm (CAN)

- Lyssna på bussen. Om bussen är ledig starta sändning av ett meddelande.
- Lyssna på bussen och jämför bit för bit av sändt data.
- Om mottagen bit skiljer sig från sändt bit indikerar detta att någon med högre prioritet skickar data samtidigt. Avsluta då sändningen och försök igen när "högprioritetsmeddelandet" har skickats i sin helhet.

•Outage time

•Kort svarstid (200  $\mu$ s vid 1Mbit)

MOP

11

## Bussåtkomst forts

### CSMA/CR (CAN)

- Carrier Sense: Lyssna först
- Multiple Access: Vem som helst kan skicka ett meddelande när bussen är ledig.
- Collision Resolution: Buss-krocksupplösning.

•Algoritm: Starkast vinner

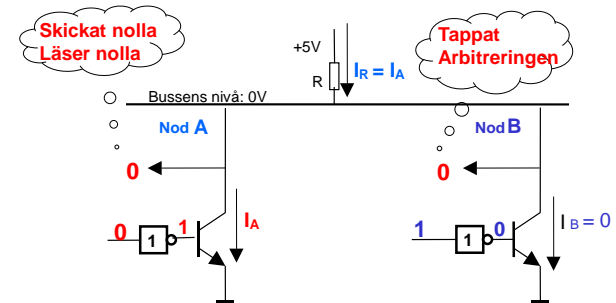
•Multimaster

MOP

10

## Bussåtkomst forts

- Bussdrivare: Princip "Open collector"
- Bussens nivå: Recessiv (bit) och Dominant (bit)



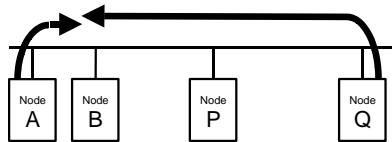
MOP

12

## Bussåtkomst forts

### Inskränkningar

- Bitsynkronisering
- En enskild bit måste vara ”giltig i hela nätet”



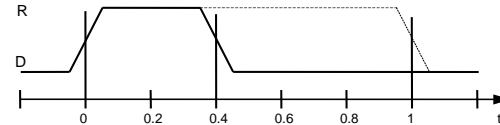
MOP

13

## Bussåtkomst forts

### Inskränkningar

- Nätets utsträckning/bithastighet



- $\tau_{vänta} = 2 l/v = 2 \cdot 40 \text{ m} / 2 \cdot 10^8 = 0,4 \mu\text{s}$
- Max 40m vid 1 Mbit

14

## Bussåtkomst

(arbitrering – tävlan om bussen)

### Tre noder skickar samtidigt

Nod A sänder: \$257 (0010 0101 0111)  
 Nod B sänder: \$360 (0011 0110 0000)  
 Nod C sänder: \$25F (0010 0101 1111)

### Bussens nivå

Bit nummer	SOF	1	2	3	4	5	6	7	8	9	10	11	12	13
Bussens nivå	D	D	D	R	D	D	R	D	R	D	R	R	R	R
Nod A skickar	0	0	0	1	0	0	1	0	1	0	1	1	1	1
Nod B skickar	0	0	0	1	1	Slutar sända								
Nod C skickar	0	0	0	1	0	0	1	0	1	1	Slutar sända			

R: Recessiv (bit)  
 D: Dominant (bit)

MOP

15

## Bussåtkomst (arbitrering)

- Identifierare anger prioriteten
- **OBS!!!** Endast EN nod får skicka data med en speciell identifierare

MOP

16

## Synkronisering i CAN

- Asynkront protokoll
- Okodade bitar

### • Synkroniseringsmekanism

- Mottagare:
  - Hård synkronisering på SOF
  - Synkronisera på flanker i mottaget data
- Sändaren:
  - Inför STUFF BITAR vid långa sekvenser av nollor / ettor

MOP

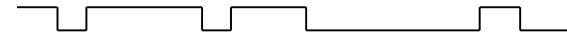
17

## Synkronisering i CAN forts

### • Stuffbitar

- Skickar ett meddelande som börjar med \$FF03 (binärt 1111 1111 0000 0011)

Bit nr	SOF	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Data	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Stuffbit							0									1		
Skickad sekv	0	1	1	1	1	1	0	1	1	1	0	0	0	0	0	1	0	0



MOP

18

## Olika ramar

- Data Frame (skicka data)
  - SOF Start Of Frame. 1 bit
  - ARB Arbitration. 12 bitar(Stand CAN)
  - CTRL Control. 6 bitar
  - DATA Användar data. 1-8 bitar (max 64 b)
  - CRC Kontrollsumma 16 bitar
  - ACK Kvitto 2 bitar
  - EOF End Of Frame 7 bitar **Totalt 108 b**
  - IFS Inter Frame Space. 3 bitar



MOP

19

## Olika ramar forts

### •Remote Frame (efterfråga en dataframe)

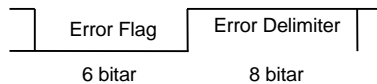
- En remoteframe saknar datafält (se data-frame).
- Remote- och dataframe har samma identifierare!!!!
  - Ex efterfråga hastighet; skicka hastighet
- ID är 11 bitar, ARB-fältet är 12 bitar
- RTR-bit (bit 12 i ARB-fältet)
  - Remote frame recessiv RTR-bit
  - Data frame dominant RTR-bit
- Innebär att vem som helst kan skicka en remoteframe (men endast EN kan skicka en data-frame)

MOP

20

## Olika ramar forts

- Error frame (skriv sönder pågående utskick)
  - 6 st dominanta bitar (Error flag)
  - 7 recessiva bitar (Error delimiter)



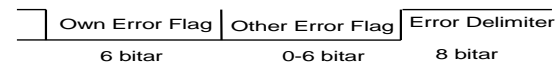
- Alla kan skicka en Error frame
- Automatisk omsändning av störda överföringar
- Robust

MOP

21

## Olika ramar forts

- Error frame forts
- Vem detekterar en error frame?
  - Sändaren av data/remote-frame
    - Utskickad bit  $\neq$  Läst bit
  - Mottagare
    - Felupptäckt



MOP

22

## Feldetektering

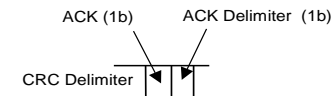
- Skicka Error Frame vid felupptäckt
  - Bit-fel
  - Stuff-fel
  - CRC-fel
  - Fixed Form-fel
- Möjliggör samtidig exekvering i noder
- Atomic broadcast

MOP

23

## Kvitto (ACK)

- En frame innehåller ett två-bitars ACK-fält för handskakning
- Sändaren av en frame skickar två recessiva bitar i detta fält
- En adresserad fungerande mottagare skriver över den första med en dominant bit
- Sändaren detekterar dominant eller recessiv bit i ACK-fältet



MOP

24

## Kvitto forts

- Vad innebär ACK-funktionen?
  - Sändaren “ser” att hans frame tas emot
  - Sändaren detekterar att den inte är “ensam i nätet”
  - Observera att sändaren inte kan detektera *vem* som korrekt tagit emot utskickad meddelande utan endast att *någon* korrekt tagit emot utskickad meddelande.
- **Underlättar konsistens**

MOP

25

## Identifierare

- Identifieraren “identifierar” meddelandet
  - ID x kan endast skickas av en viss nod pga arbitrering
- Standard CAN
  - 11 bitar- 2048 olika identifierare
- Enskilda meddelanden
  - **\$04B** Bromskommando
  - **\$1A2** Ljus meddelande, fram

MOP

26

## Identifierare forts

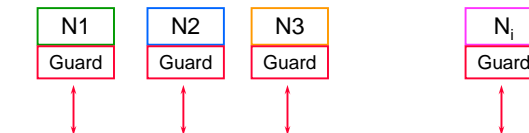
- Extended CAN (29 bitar -  $0.5 \cdot 10^9$  identifierare)
- Försök till standardisering (anpassning J1939)
  - Standardiserade identifierare
  - **\$1A29 92D0** HF parklampa
  - **\$1A29 22D1** HF halvljuslampa
  - **\$1A29 32D2** HF helljuslampa

MOP

27

## Identifierare forts

- Filtrering
  - En CAN-krets initieras (i mask-registret) med ett antal identifierare som den skall igenkänna



- **Multicast**
- **Belastar ej processorn med önskad data**

MOP

28

## Identifierare forts

- Filtrering och prioriteter
  - \$1A29 92D0 HF parklampa
  - \$1A29 22D1 HF halvljuslampa
  - \$1A25 32D2 HF helljuslampa
  - \$FFF0 0FF0 Maskregister
- Möjliggör dynamisk prioritetsändring
- Outage times

MOP

29

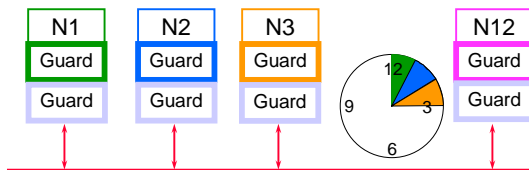
## Sammanfattning CAN

- Flexibelt
- Multimaster protokoll
- Multicast protokoll
- Hög överföringshastighet (1 Mbit/s)
- Kort svarstid (200 µs)
- Automatisk omsändning av störda överföringar
- "Atomic Broadcast"
- Stöd för synkroniserad exekvering i olika noder
- Avlastar processorn/användaren med meddelandeöverföringen

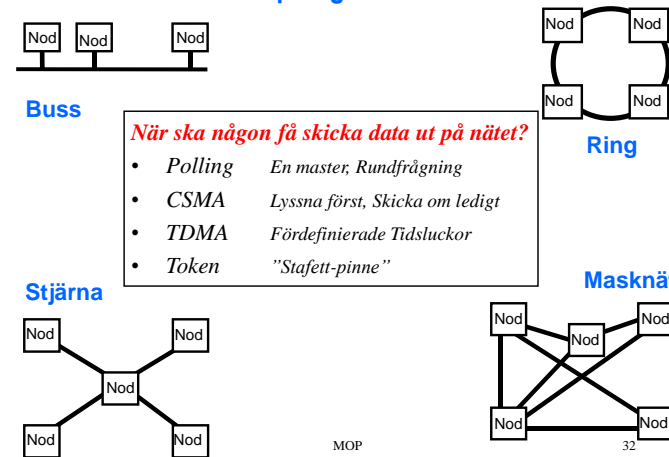
MOP

30

## Uppfyller CAN tidkraven?



## Datanät- Topologi - Accessmetoder



MOP

32