Exercise 6 - Notes

Mittwoch, 13. Juli 2022 18:04

Exercise (SS 2022)

Comunication Systems and Protocols

Institut fuer Technik der Informationsverarbeitung
Dr. Jens Becker, Matthias Stammler M. Sc.

Task 1: CAN Bus

/4

Since CAN uses CSMA/CR as arbitration scheme every participant compares the actual bus level with the signal transmitted by itself. Because of that it is important for every participant to be able to evaluate the actual state on the bus before begin of a new bit. Here beside the signal runtime on the bus also the required processing time of the participant itself plays a role.

As given in Figure 1.1, this includes the processing time t_{CAN} of the CAN controller, the times t_{Rx} and t_{Tx} which are needed inside the transceiver for reception and transmission as well as the runtime t_{Bus} on the bus.

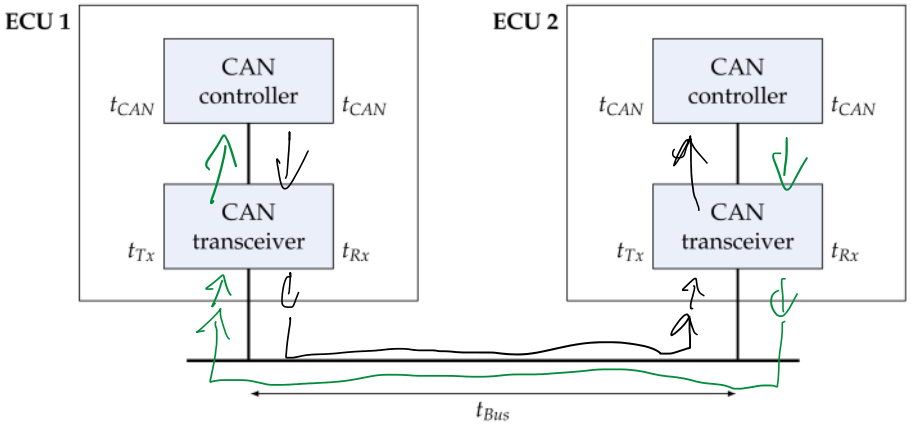

Figure 1.1: CAN bus

1.1 What is the interrelation between the maximum bus length and the bit transmission rate for CAN? Neglect the processing time inside the ECUs for this question

/1

1.2 Based on the previous question, specify the maximum bus length for a speed of propagation of $v = 2.3 \cdot 10^8 m/s$ and for the transmission rates of $10 kbit/s$ and $1 Mbit/s$ respectively.

/1

1.3 Now also consider the delays inside the ECUs. What is the maximum data transmission rate if the bus length between the two furthest controllers is 300 meters? The detection of the bus state shall be accomplished after 80% of the bit time at latest. (assume: $t_{CAN} = 75 nsec$, $t_{Rx} = t_{Tx} = 25 nsec$, $v_{Bus} = 0,2 m/nsec$).

/2

1.1: T_{Bit}: Transmission-Duration
T_{prop}: Runtime on Bus

$\frac{1}{TR} \overset{!}{\gg} \frac{l}{v}$ $\quad T_{Bit} \gg T_{prop}$

1.2: $T_{Bit} \overset{!}{\geq} 2 \cdot T_{prop}$ ← T_{Bit} has to be twice as big because of travel-time from node 1 to node 2 and back.

put in values: $\frac{1}{TR} = 10\,kBit/s \rightarrow l \leq \underline{\underline{11500\,m}}$

$\frac{1}{TR} = 1\,MBit/s \rightarrow l \leq \underline{\underline{115\,m}}$

1.3: $t_{TN} = t_{con} + t_{tx} + t_{con} + t_{rx} = 2 \cdot 75\,\mu s + 2 \cdot 25\,\mu s$

$= 200\,\mu s$

$t_{Bus} = \frac{l_{Bus}}{v_{Bus}} = \frac{300\,m}{0.2\,m/\mu s} = 1.5\,\mu s$

Min. Bit Duration $t_{Bit} \geq 2 \cdot (t_{TN} + t_{Bus}) = 3.4\,\mu s$

$t_{Bit,buffered} = \frac{t_{Bit}}{0.8}$ $\quad S = \frac{1}{t_{Bit,buffered}} = \frac{0.8}{3.4\,\mu s} = \underline{\underline{235\,kBit/s}}$

Task 2: Universal Serial Bus (USB)

/2

2.1 Consider an USB 1.1 device in reset state. Calculate the current on the bus. Neglect the energy needs of the device itself. Use the circuit in Figure 2.1 as orientation. The current flow on the bus is determined by the series connection of R_1 and R_2:

/1

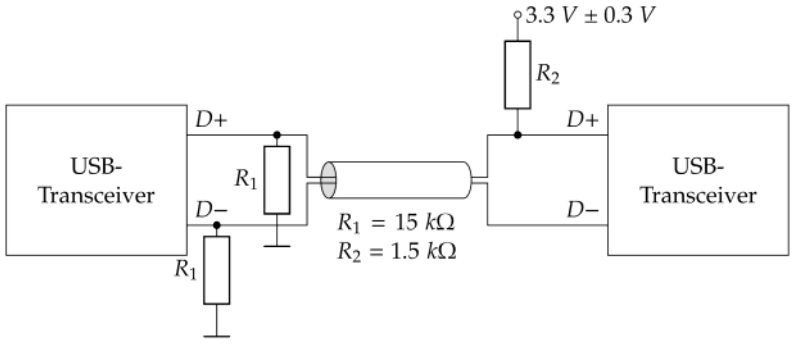

Figure 2.1: Example for resistor configuration at 12Mbit/s

2.2 To reduce the probability of errors during the handshake phase of a transaction, two data PIDs (DATA0 and DATA1) are used. The data PID is changed after every successful transmission. For that reason the sender and the receiver both have a „data toggle sequence bit“. At the receiver this only changes if correct data with a correct PID has been accepted. At the sender it changes when a valid ACK-Handshake is received. Both participants of a transmission first have to synchronize their bits during the setup phase of a control transfer (see Figure 2.2, the bracketed values correspond to the value of the “data toggle sequence bits”; at X/Y they are still undefined).
Starting from the state of 2.2, specify the flow charts for the following cases (always consider a transfer from host to device):

/1

1. A successful transmission
2. A data packet is rejected and only accepted after being transmitted again
3. The handshake packet of a transmission has been mutilated. The data is transmitted again and accepted then

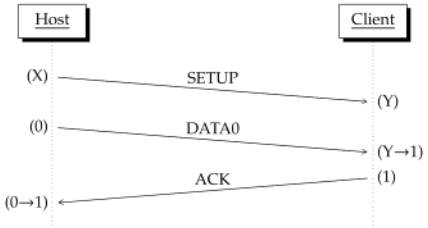

Figure 2.2: Synchronization of the data toggle sequence bits

4 SETUP, OUT and in are different token packets.

Please refer to solution sheet for nice drawings.

Matr. №: ID:

Task 3: Routing - old exam question

/12

Switching

/3

3.1 Figure 3.1 shows a network topology with equal weights w_n on every link. Give the number of hops of the route from node A to F using *minimal routing*. In addition, describe how the number of hops from A to F could change if weights are differing and non-minimal routing is used.

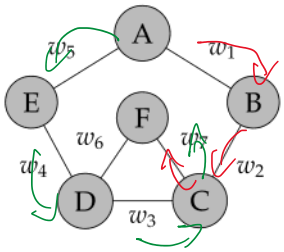

Figure 3.1: A network topology

A -> F : 3 hops

A -> F C non-minimal : 4 hops

/9

3.2 Figure 3.2 represents a network for which an optimal routing has to be found. The weights represent an abstract metric for traffic present at each connection. With node **A** as the starting point, calculate the paths with the lowest total traffic in the network by using Dijkstra's algorithm. Fill in Table 3.1 with the order in which nodes are visited in each step and the shortest paths after visiting the corresponding node. Make sure to fill all fields of Table 3.1.

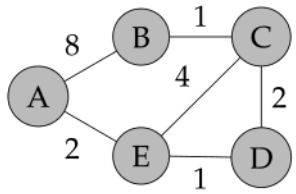

Figure 3.2: Given network topology

	step 0		step 1		step 2		step 3		step 4		step 5	
node	**A**		A		E		D		C		B	
vertex	dist.	pred.	dist.	pred.	dist.	pred.	dist.	pred.	dist.	pred.	dist.	pred.
A	0	A	0	A	0	A	0	A	0	A	0	A
B	∞	-	8	A	8	A	8	A	6	C	6	C
C	∞	-	9	-	6	E	5	D	5	D	5	D
D	∞	-	∞	-	3	E	3	E	3	E	3	E
E	∞	-	2	A	2	A	2	A	2	A	2	A

Table 3.1: Dijkstra's algorithm