

Feldbusse und Sensornetze

Übungsblatt 14/06/2005

Aufgabe1: CAN theoretisch:

- a) Welche Mechanismen der Fehlererkennung sind im CAN-Bus Protokoll enthalten?
- b) Wie setzt CAN das alles oder nichts Prinzip der Nachrichtenübertragung durch?
- c) Welche Garantie kann der CAN-Bus für die Versendung bzw. den Empfang von Nachrichten geben?
- d) Mit welchen Mechanismen kann man CAN-Nachrichten bereits vom Controller aussieben lassen und wie ist deren Funktionsweise?
- e) Wodurch können Nachrichten-Duplikate entstehen?
- f) Können Nachrichten-Duplikate vom CAN-Protokoll verhindert werden?
- g) Eine Koexistenz von TT- und ET-Kommunikation ist möglich. Wie funktioniert sie und wie erfolgt die Sicherstellung der TT-Eigenschaften?
- h) Können zwei Nachrichten die gleiche Deadline besitzen und trotzdem ohne Verletzung der TT-Eigenschaft propagiert werden und wenn ja wie und welche würde im Bedarfsfall bevorzugt?

Aufgabe 2: CAN praktisch:

Gegeben ist ein System mit 4 CAN-Knoten (Linux-Workstations).

1. Schreiben Sie ein Programm, welches auf dem CAN-Bus laufende Nachrichten auf dem Bildschirm ausgibt.
2. Schreiben Sie ein Programm, welches Nachrichten auf dem CAN-Bus in bestimmbaren Zeitintervallen sendet.
3. Schreiben Sie ein Programm, dass Filtermöglichkeiten bietet und beispielsweise nur die Nachrichten des Types 0x400 aus unten stehender Tabelle ausgibt.

- a) Konfigurieren Sie das System auf eine BAUD-Rate von 250 kbps.
- b) Senden sie die folgenden Nachrichten nach dem aufgeführten Schema

ID	DATA	Interval
0x400	0x01 0x02 0x03 0x04	1 s
0x300	0x0a 0x0b 0x0c 0x0d	0.5 s
0x200	0xf0 0xf0 0xf0 0xf0	0.2 s
0x100	0xff 0xff	0.1 s

Hinweis:

**Im Labor besteht ein CAN-Knotennetz zwischen den Rechnern (EOS-LAB 1-4).
Zum Testen des eigenen Monitor programms sowie zum Testen des eigenen
Sendeprogramms stehen die Programme receivetest und transmittest zur
Verfügung.**

Programmierungsumgebung:

- 1. include-Dateien für die Verwendung der CAN-Schnittstelle sind pcan.h sowie libpcan.h**
- 2. Die zuzubindende Bibliothek ist libpcan.so.**
- 3. Das zu verwendende Device ist /dev/pcan24**
- 4. mit Hilfe von cat /proc/pcan kann der Status der CAN-Knotens aus der Shell heraus abgefragt werden**
- 5. für das Programm transmittest steht unter /home1/mschulze/transmit.txt eine Sendedatei zur Verfügung**