



---

Übungsblatt 7

Abgabetermin 20.06.2006

---

**Aufgabe 1: CAN theoretisch**

- a) Worauf beruht die Priorisierung der Nachrichten im CAN Arbitrierungsverfahren und wie funktioniert dieses Verfahren?
- b) Welche Garantie kann das CAN-Protokoll für die Versendung bzw. den Empfang von Nachrichten geben?
- c) Wovon hängt die Buslänge bei CAN ab und warum?
- d) Dürfen zwei Nachrichten mit der gleichen ID von zwei verschiedenen Knoten auf den CAN-Bus gelegt werden? Begründen Sie Ihre Aussage.
- e) Es gibt die CAN-Spezifikation 1.2 und 2.0. Was war der Grund für die Einführung der Spezifikation 2.0? Welches sind die grundlegenden Unterschiede zwischen den Spezifikationen?
- f) Beschreiben Sie die verschiedenen Nachrichtentypen (Aufbau, Einsatzzweck) des CAN-Protokolls.

**Aufgabe 2: CAN praktisch**

Gegeben ist ein System mit 4 CAN-Knoten (Linux-Workstations). Das CAN-Netz soll mit einer BAUD-Rate von  $250\text{kbps}$  betrieben werden..

- a) Schreiben Sie ein Programm, welches auf dem CAN-Bus laufende Nachrichten auf dem Bildschirm ausgibt.
- b) Schreiben Sie ein Programm, welches Nachrichten auf dem CAN-Bus in bestimmten Zeitintervallen (siehe Tabelle 1) sendet

Die praktischen Aufgaben werden im Anschluss an die Besprechung der theoretischen Aufgaben von den Studenten am Rechner vorgestellt und durch Sebastian oder Michael abgenommen.

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

Tabelle 1: Nachrichtenspezifikation

### Hinweis

Im Labor besteht ein CAN-Knotennetz zwischen den Rechnern (EOS-LAB 1-4) auf der rechten Seite. Zum Testen des eigenen Monitorprogrammes sowie zum Testen des eigenen Sendeprogramms stehen die Programme `receivetest` und `transmittest` unter `/usr/local/bin/` zur Verfügung.

### Programmierungsumgebung:

- include-Dateien für die Verwendung der CAN-Schnittstelle sind **pcan.h** sowie **libpcan.h**.
- Die zuzubindende Bibliothek ist **libpcan.so**.
- Das zu verwendende Device ist **/dev/pcan24**.
- Mit Hilfe von **cat /proc/pcan** kann der Status des CAN-Knotens aus der Shell heraus abgefragt werden
- Um den PCAN-Dongle neu zu initialisieren, ist es nicht notwendig den Rechner neu zu starten. Mit dem Befehl **sudo /etc/init.d/pcan restart** wird die Reinitialisierung durchgeführt.
- Für das Programm `transmittest` steht unter `/home/mschulze/transmit.txt` eine Datei mit Nachrichtendefinitionen zum Senden zur Verfügung.

### Punkteverteilung

Aufgabe	Punkte
CAN theoretisch	3 / 1 / 2 / 3 / 3 / 8
CAN praktisch	x / x