



Übungsblatt 8

Abgabetermin 27.06.2006

Aufgabe 1: CAN theoretisch

- a) Welche Mechanismen der Fehlererkennung sind im CAN-Bus Protokoll enthalten?
- b) Wie setzt der CAN das alles oder nichts Prinzip der Nachrichtenübertragung durch?
- c) Mit welchen Mechanismen kann man CAN-Nachrichten (CAN 2.0) bereits vom Controller aussieben lassen und zeigen Sie an einem selbstgewählten Beispiel die Funktionsweise?
- d) Eine Koexistenz von TT- und ET-Kommunikation ist möglich. Wie funktioniert sie und wie erfolgt die Sicherstellung der TT-Eigenschaft?
- e) Können zwei Nachrichten die gleiche Deadline besitzen und trotzdem ohne Verletzung der TT-Eigenschaft propagiert werden. Wenn ja wie und weiterhin welche Nachricht wird im Bedarfsfall bevorzugt?
- f) Gegeben ist ein CAN-Bus mit einer Übertragungsbandbreite von 1MBit/s , bei dem die Nachrichtenübertragung stets ohne Fehertelegramme läuft. Berechnen Sie die maximal mögliche Nachrichtenaufkommen für
 - 1) CAN-Spezifikation 1.2 und einem Data-Frame mit 8 Byte Payload
 - 2) CAN-Spezifikation 2.0 und einem Data-Frame mit 8 Byte Payload

Aufgabe 2: CAN praktisch

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

- a) Schreiben Sie ein Programm, das Filtermöglichkeiten bietet und beispielsweise nur die Nachrichten mit der ID $0x400$ auf dem Bildschirm ausgibt.

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-Knotenetz zwischen den Rechnern (eoslab-0{1,3,4,6}) 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	5 / 2 / 4 / 3 / 2 / 4
CAN praktisch	x