



Übungsblatt 02

Allgemeine Hinweise

Ab diesem Übungsblatt werden Punkte vergeben. Um eine Scheinprüfung am Ende des Semesters ablegen zu können, sind 50% der Übungspunkte und eine regelmäßige Teilnahme an der Übung nötig, in der auch eine eigene Lösung vorgestellt werden muss. In die Bewertung von Programmieraufgaben fließt auch die sinnvolle Kommentierung der Quelltexte mit ein.

Die Teilnahme an der ersten Übungsstunde am 25.04.2005 ist dringend erforderlich, da dort die Organisation und das durchgehende Beispiel der weiteren Übungsblätter vorgestellt wird.

Implementierung eines CAN Treiber Stubs

Ein Teil der Übungsaufgaben ergibt am Ende der Vorlesung ein zusammenhängendes Beispiel aus dem Bereich Automobilbau. Es soll ein Dynamisches Stabilitätsprogramm (DSP) entwickelt werden, das Anti-Blockier-System (ABS), Antriebschlupfregelung (ASR) und Schleudervermeidung realisiert. An der Funktion sind verschiedene Steuereinheiten beteiligt, die alle über ein *Controller Area Network* (CAN) kommunizieren.

Implementieren Sie ein Ada Paket `can_driver` mit einem Stub für einen CAN-Treiber.

Aufgabe 3 Datentyp für CAN Nachrichten (3 Punkte)

CAN Nachrichten (`can_msg_transmit_struct`) haben eine Integer ID (`id_raw`), eine Integer Angabe über die Länge der Daten (`dlc_raw`, *Data Length Code*) und ein Datenfeld für acht Integer-Werte (`data_fld`), Index beginnend bei 0.

Implementieren Sie diesen Datentyp.

Aufgabe 4 Debug-Ausgabe für CAN Nachrichten (3 Punkte)

Implementieren Sie eine Prozedur `dbg_message` für die Debug-Ausgabe von CAN Nachrichten, die eine Nachricht vom Typ `can_msg_transmit_struct` als Parameter erhält und in folgender Form ausgibt:

```
*** CAN Message ***   ID: <id_raw>   Data: <data_fld(0)> <data_fld(1)> ...
```

Dabei sollen auch nur so viele Datenfelder ausgegeben werden, wie in `dlc_raw` vermerkt sind.

Aufgabe 5 Übertragungsstub (2 Punkte)

Ergänzen Sie den CAN Treiber um eine Funktion `transmit`, die eine Nachricht vom Typ `can_msg_transmit_struct` als Parameter erhält, eine Debug-Ausgabe für diese Nachricht erzeugt und einen Integer-Wert für den Status zurückgibt. `k_can_tx_ok` (=0) steht dabei für eine korrekte Übertragung `k_can_tx_failed` (=1) für einen Fehler bei der Übertragung.

Aufgabe 6 Testprozedur für CAN Treiber (2 Punkte)

Implementieren Sie eine Testprozedur `test_can_driver`, die zehn unterschiedliche Nachrichten mit dem CAN-Treiber übermittelt.

Abgabe: Do 28.04.2005

Sie können Ihre Lösung zum obigen Termin in Papierform in der Vorlesung oder in elektronischer Form per eMail an ada@i3.informatik.rwth-aachen.de abgeben. Bei Programmieraufgaben senden Sie bitte die Quellcode-Dateien in jedem Fall an die o.g. eMail-Adresse.

Bitte vermerken Sie in jedem Fall die Namen und Matrikelnummern aller beteiligten Personen (maximal 3), beim Versenden per eMail auch alle eMail-Adressen. Abgaben in elektronischer Form können ausschließlich in den Dateiformaten *Plain-Text* oder *PDF* erfolgen.

Aktuelle Informationen zur Vorlesung finden Sie auf den Webseiten des Lehrstuhls unter <http://www-i3.informatik.rwth-aachen.de>