

**Laborversuch für die Lehrveranstaltung:  
Kommunikationssysteme in Straßenfahrzeugen (KIS) / Prof Dr.-Ing. Kabulepa**

CAN Datenübertragung mit dem Mikrocontroller LPC2368

Laborversuch: **CAN Bussysteme**

Name, Vorname:

Matr.-Nr.:

WS/SS \_\_\_\_\_

Gruppennummer: \_\_\_\_\_

Testat:

## Teil 1: Versuchsvorbereitung

### ➤ Aufgabe 1

Zur Einstellung der CAN Datenrate sollen verschiedene Werte fürs BTR (Bus Timing) Register in der Datei **emb1\_labs.h** vorbereitet werden.  
Passen Sie bitte die Werte der unterstehenden Parameter in der Datei **emb1\_labs.h** angesichts der zu wählenden Peripherie-Frequenz und Übertragungsdatenrate an.

```
#define BITRATE100K14_4MHZ 0x00000000 //100 Kbaud @14,4MHz PeripherieClock
#define BITRATE125K14_4MHZ 0x00000000 //125 Kbaud @14,4MHz PeripherieClock
#define BITRATE250K14_4MHZ 0x00000000 //250 Kbaud @14,4MHz PeripherieClock
#define BITRATE500K14_4MHZ 0x00000000 //500 Kbaud @14,4MHz PeripherieClock
#define BITRATE1M14_4MHZ 0x00000000 //1000 Kbaud @14,4MHz PeripherieClock

#define BITRATE250K28_8MHZ 0x00000000 //250 Kbaud @28,8MHz PeripherieClock
```

## ➤ Aufgabe 2

Ein CAN-Knoten ist an keinem Bussystem angebunden. Welche Werte soll man in folgenden Registern bzw. Bitfeldern von Registern erwarten, nachdem der Knoten Daten gesendet hat.

- TXERR (Tx Error Counter) im Register CANxGSR (x = 1 oder 2)
- Rx Buffers (CANxRFS, CANxRID, CANxRDA und CANxRDB)  
(x = 1 oder 2 je nachdem, ob CAN Controller 1 oder 2 verwendet wird)

Folgende Betriebsmodi werden in Betracht gezogen:

- a) Normaler Betriebsmodus
- b) Normaler Betriebsmodus mit gesetztem AT (Abort Transmission) Bit beim Senden.
- c) Globaler Self-Test
- d) Globaler Self-Test mit mit gesetztem AT (Abort Transmission) Bit beim Senden.
- e) Lokaler Self-Test

## Teil 2: Versuchsdurchführung

### ➤ Aufgabe 3: Single-Node Test

Verwenden Sie die gelieferten CAN API Funktionen, um den CAN Controller 1 in den folgenden Betriebsmodi zu verwenden.

- a) Normaler Betriebsmodus
- b) Normaler Betriebsmodus mit gesetztem AT (Abort Transmission) Bit beim Senden.
- c) Globaler Self-Test
- d) Globaler Self-Test mit mit gesetztem AT (Abort Transmission) Bit beim Senden.
- e) Lokaler Self-Test

Beobachten Sie die Änderungen im Bitfeld TXERR (Tx Error Counter) des Registers CAN1GSR sowie in den CAN1 Rx Buffers (CAN1RFS, CAN1RID, CAN1RDA und CAN1RDB).

Überprüfen Sie bitte, ob die beobachteten Änderungen Ihren Erwartungen entsprechen.

#### ➤ Aufgabe 4: Single-Node CAN1-CAN2 Loopback

Nun sollen die zwei CAN Controller des Mikrocontrollers LPC2368 über den externen CAN Bus Daten austauschen. Die Verbindung wird mit den Anschüssen CANH und CANL realisiert. Zwischen den Leitungen CANH und CANL soll ein Terminierungswiderstand angeschlossen sein.

→ Überprüfen Sie, ob Sie Daten senden und empfangen können. Beide Richtungen sollen überprüft werden. (Richtung 1: CAN1 sendet, CAN2 empfängt / Richtung 2: CAN 2 sendet, CAN1 empfängt).

Fall 1: Datenrate 250 Kbaud, Telegrammlänge 4 Bytes

Fall 2: Datenrate 1 Mbaud, Telegrammlänge 8 Bytes

→ Messen Sie die Übertragungszeit von einem 2-KByte Datenpaket.

Fall 1: Datenrate 250 Kbaud, Telegrammlänge 4 Bytes

Fall 2: Datenrate 1 Mbaud, Telegrammlänge 8 Bytes

Hinweis: Für die Zeitmessung können Sie ein Oszilloskop verwendet, um Zeitintervalle eines GPIO (General-Purpose Input Output) zu erfassen. Als Alternative kann auch ein interner Timer von Mikrocontroller verwendet werden. Weitere Hinweise hierzu erhalten Sie während der Versuchsdurchführung.

#### ➤ Aufgabe 4: Single-Node CAN1-CAN2 Loopback (Fortsetzung)

