

CAN Treiber

Generated by Doxygen 1.8.13

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

CANDriver	
CAN Treiber	??

Chapter 2

Class Documentation

2.1 CANDriver Class Reference

CAN Treiber.

```
#include "CAN/CANDriver.hpp"
```

Public Member Functions

- void [init](#) (CAN0Rx rx_pin, CAN0Tx tx_pin, uint32_t baudrate)
CAN0 Initialisierungsfunktion.
- void [init](#) (CAN1Rx rx_pin, CAN1Tx tx_pin, uint32_t baudrate)
CAN1 Initialisierungsfunktion.
- bool [checkStatusInterrupts](#) (void)
Ueberprueft den Status des Treibers.
- uint32_t [getMessageInterruptMask](#) (void)
Checkt ob neue Nachrichten empfangen wurden.
- tCANMsgObject [getMessage](#) (uint32_t postbox)
Gibt Inhalt der als Parameter uebergebenen Postbox zurueck.

Protected Member Functions

- void [sendCANMessage](#) (tCANMsgObject *txCANMsg, uint32_t postbox)
Funktion zum senden von CAN Nachrichten.
- virtual void [registerReceiveMessages](#) (void)
Registrieren der CAN Nachrichten, die gelesen werden sollen.

Protected Attributes

- tCANMsgObject **txMessage**
- CAN_BASE **can_base**
- tCANMsgObject **rxMessage**

2.1.1 Detailed Description

CAN Treiber.

Diese Klasse kann als Treiber fuer die CAN Kommunikation in Verbindung mit dem TM4C123GH6PM genutzt werden. Es stehen 32 Postboxen zur Verfuegung. Von dieser Klasse sollte immer eine Tochterklasse erstellt werden und damit gearbeitet werden. Bitte diese Klasse nicht editieren oder nur nach Absprache mit STL.

2.1.2 Member Function Documentation

2.1.2.1 checkStatusInterrupts()

```
bool CANDriver::checkStatusInterrupts (
    void )
```

Ueberprueft den Status des Treibers.

Ueberprueft ob Fehler vorliegen und ob das Modul in den passiven Modus gewechselt ist(es koennen keine CAN Nachrichten mehr empfangen und gesendet werden). Liegt dieser Fall vor, wird automatisch wieder in den aktiven Modus gewechselt. Sollte immer vor [getMessageInterruptMask\(\)](#) aufgerufen werden.

2.1.2.2 getMessageInterruptMask()

```
uint32_t CANDriver::getMessageInterruptMask (
    void )
```

Checkt ob neue Nachrichten empfangen wurden.

Gibt eine Mask zurück, die angibt, in welcher Postbox ein Interrupt aufgetreten ist. Diese Nachricht kann danach mit [getMessage\(\)](#) gelesen werden. Zuvor sollte immer [checkStatusInterrupts\(\)](#) ausgefuehrt werden.

2.1.2.3 init() [1/2]

```
void CANDriver::init (
    CANORx rx_pin,
    CANOTx tx_pin,
    uint32_t baudrate )
```

CAN0 Initialisierungsfunktion.

Initialisiert den Treiber fuer die uebergebenen Pins und Baudrate. Die Pins werden komplett konfiguriert, ggf. Hardware entlockt. Es wird automatisch an der Auswahl der Pins erkannt, welcher der beiden internen CAN-↔ Module genutzt wird.

2.1.2.4 init() [2/2]

```
void CANDriver::init (
    CAN1Rx rx_pin,
    CAN1Tx tx_pin,
    uint32_t baudrate )
```

CAN1 Initialisierungsfunktion.

Initialisiert den Treiber fuer die uebergebenen Pins und Baudrate Die Pins werden komplett konfiguriert, ggf. Hardware entlockt. Es wird automatisch an der Auswahl der Pins erkannt, welcher der beiden internen CAN-Module genutzt wird.

2.1.2.5 registerReceiveMessages()

```
void CANDriver::registerReceiveMessages (
    void ) [protected], [virtual]
```

Registrieren der CAN Nachrichten, die gelesen werden sollen.

Diese Funktion ist hier leer und sollte zur Benutzung in der Tochterklasse ueberschrieben werden. Pro Nachricht die empfangen werden soll wird eine Postbox benoetigt. Insgesamt gibt es 32 Postboxen. Es kann keine Postbox zum Senden und Empfangen genutzt werden.

2.1.2.6 sendCANMessage()

```
void CANDriver::sendCANMessage (
    tCANMsgObject * txCANMsg,
    uint32_t postbox ) [protected]
```

Funktion zum senden von CAN Nachrichten.

Mit dieser Funktion kann eine Nachricht in eine Postbox platziert werden, die dann automatisch versendet wird. Wenn moeglich, sollte eine Postbox jeweils nur fuer eine Nachricht genutzt werden, um nicht gewuenschte Ueberschreibungen in der Postbox zu vermeiden. Auch darf keine Postbox gewaehlt werden, die bereits zum Empfangen von Nachrichten genutzt wird. Insgesamt gibt es 32 Postboxen.

The documentation for this class was generated from the following files:

- D:/KaRaceing/Software/KIT18/TIDriverlib/CAN/CANDriver.hpp
- D:/KaRaceing/Software/KIT18/TIDriverlib/CAN/CANDriver.cpp

