Public Support - Support Request #626

ADTFS-46726 Provide CAN-FD structure

2017-05-10 13:30 - hidden

Status:	Closed		
Priority:	Normal		
Category:			
Customer:	ELEKTROBIT	**Product Issue Numbers:**	
Department:	SUPPORT	**Affected Products:**	ADTF 2.14.0, ADTF 3.0.0, ADTF Device Toolbox 2.7.0
Requester's Priority:	Normal	**Platform:**	Ubuntu 12.04 64bit, Ubuntu 16.04 64bit, Windows 7 64bit
Support Level:	2nd Level	**Topic:**	DeviceTB::CANFD
Resolution:	Solved Issue	**FAQ Links:**	

Description

Supportanfrage:

Wir haben aktuell eine ganze Reihe von Anfragen bzgl. CAN-FD und ich bin gerade am überlegen, ob wir das für ADTF-2 in unsere ARXML-Toolbox integrieren.
Um hier kompatible zu sein mit
a) der VW-Internen Implementierung und
b) mit der Implementierung, die im Laufe des Jahres in ADTF-3 integriert werden soll wollte ich fragen, ob ihr die Datenstruktur mit uns teilen könnt. In der aktuellen Beta der Device-TB ist ja noch nichts in der Richtung zu finden (auf jeden Fall nicht in der Doku).

Grüße,
Simon

Best regards - Beste Grüße
Ursula Groh

EB Assist ADTF Support-Team

Lösung:

Die Bereitstellung der Struktur erfolgt laut KuFo Beschluss im Rahmen des Feature Complete.
Die Integration von CAN-FD in DevTB 3.x ebenso, derzeit auf 2017b geplant.

Siehe canfd_types.h.

History

#1 - 2017-05-10 15:40 - hidden

- Project changed from Public Support to 7

- Description updated

- Status changed from New to In Progress

- Topic set to DeviceTB::CANFD

- Support Level changed from 2nd Level to 3rd Level

- Customer set to ELEKTROBIT

- Department set to SUPPORT

- Affected Products ADTF 2.14.0, ADTF 3.0.0, ADTF Device Toolbox 2.7.0 added

- Platform Ubuntu 12.04 64bit, Ubuntu 16.04 64bit, Windows 7 64bit added

#2 - 2017-05-11 09:33 - hidden

@Stephan:

- Bitte CAN-FD Struktur aus DevTB 2.7 hier im Ticket bekannt machen, Ticket bleibt private (!). Damit kann EB arbeiten.

@EB:

- Die Bereitstellung der Struktur erfolgt laut KuFo Beschluss im Rahmen des Feature Complete.
- Die Integration von CAN-FD in DevTB 3.x ebenso, derzeit auf 2017b geplant.

#3 - 2017-05-11 09:35 - hidden

- Support Level changed from 3rd Level to 2nd Level

#5 - 2017-05-11 11:18 - hidden

- File canfd_types.h added

- Status changed from In Progress to Customer Feedback Required

- Resolution set to Solved Issue

Hallo Ursula,

hier die CAN-FD Struktur:

Die canfd_types.h liegt im Anhang.

```
struct tCANFDData
{
    /**
     *  This enum specifies the different kinds of messages
     *  that may be contained in the union
     */
    enum eMessageTag
    {
        MT_Data = 0,        //!< Data
        MT_Status = 1,      //!< Status
    };

    /**
     *  CAN FD message flags
     */
    enum eDataFlags
    {
        DF_NONE = 0                          //!< Standard flags
        , DF_ERROR_FRAME = 1             //!< Indicates an error frame
        , DF_REMOTE_FRAME = 2            //!< Indicates a remote frame
        , DF_TX_COMPLETED = 4             //!< Notification for successful message transmission
        , DF_EXTENDED_DATA_LENGTH = 8
//!< Indicates FlexibleData-Rate (FDF Flexible Datarate Format Indicator)
        , DF_CAN_FD_FORMAT_IDENTIFIER = DF_EXTENDED_DATA_LENGTH //!< same as Extended Data Length
        , DF_BAUD_RATE_SWITCH = 16      //!< Indicates that the Message uses Flexible Datarate
        , DF_SET_CAN_FD_AND_BRS = DF_CAN_FD_FORMAT_IDENTIFIER | DF_BAUD_RATE_SWITCH
//!< Helper to Set Extended Data Length and Baud Rate Switch at once
        , DF_ERROR_STATE_INDICATOR = 32
//!< EDI Bit Indicates an Error-active state at the CAN FD Node
    };

    /**
     *  CAN FD bus state flags.
     */
    enum eBusStatus
    {
        BS_OFFLINE = 1,         //!< Bus is offline
        BS_ERROR_PASSIVE = 2,   //!< One of the error counters has reached the error level.
        BS_ERROR_WARNING = 4,   //!< One of the error counters has reached the warning level.
        BS_ERROR_ACTIVE = 8     //!< Bus is online
    };

    /**
     *  CAN FD message id masks. These masks should be used to check for extended or standard messages
     *  and to get the correct identifier from ui32Id in tData.
     */
    enum eMsgId
    {
        MSG_IDMASK_BASE         = 0x000007FF   //!< Message IDs for base frame format use 11 bit identifiers
        , MSG_IDMASK_EXTENDED   = 0x1FFFFFFF
//!< Message IDs for extended frame format use 29 bit identifiers
        , MSG_EXTENDED_FLAG     = 0x80000000   //!< Extended CAN messages are marked by bit 31
    };
```

```
    /**
     *  CAN FD message header structure
     */
    struct tMessageHeader
    {
        tUInt8      ui8Tag;             //!< Type of contained message (see eMessageTag)
        tUInt8      ui8Channel;         //!< Channel that received this message
        tTimeStamp  tmTimeStamp;        //!< Hardware timestamp in micro seconds
    };

    /**
     *  CAN FD message data structure
     */
    struct tData
    {
        tUInt32     ui32Id;
//!< id of can message. For extended CAN messages bit 31 is set. Use the members of the enum eMsgId to get the
 identifier and check for extended messages.
        tUInt8      ui8Length;
//!< length of data [0..8,12,16,20,24,32,48,64] @attention DLC on Physical layer is coded differently
        tUInt8      ui8Reserved;        //!< reserved, should be zero
        tUInt16     ui16Flags;          //!< Flags @see eDataFlags
        tUInt16     ui16Reserved;       //!< reserved, should be zero
        tUInt32     ui32Reserved;       //!< reserved, should be zero
        tUInt8      aui8Data[64];       //!< data field
    };

    /**
     *  CAN FD bus status structure
     */
    struct tStatus
    {
        tUInt32     ui32BitRate;        //!< Arbitration or nominal CAN FD bus bitrate
        tUInt32     ui32RxBitCount;     //!< Count of received bits
        tUInt32     ui32TxBitCount;     //!< Count of transmitted bits
        tUInt16     ui16RxErrorCounter; //!< Error counter for the receive section of the CAN controller.
        tUInt16     ui16TxErrorCounter; //!< Error counter for the transmit section of the CAN controller.
        tUInt8      ui8BusStatus;       //!< Flags @see eBusStatus
        tUInt8      ui8Reserved;        //!< reserved, should be zero
        tUInt32     ui32DataBitRate;
//!< alternate Bitrate used for Data in CAN FD messages with BRS flag set.
        tUInt8        aui8Reserved[56];   //!< reserved, should be Zero
    };

    tMessageHeader  sHeader;            //!< CAN message header structure

    union
    {
        tData   sData;                  //!< used when ui8Tag == MT_Data
        tStatus sStatus;                //!< used when ui8Tag == MT_Status
    };
};
```

Damit sollte das Supportticket abgeschlossen sein.
Gibt es noch Fragen dazu, ansonsten bitte ein kurzes Feedback, damit wir es schließen können.

#6 - 2017-05-16 08:16 - hidden

Bitte um Feedback bis spätestens morgen den 17.05.2017.
Thema aus Supportsicht eigentlich abgeschlossen.

#7 - 2017-05-16 12:59 - hidden

Kann geschlossen werden. Vielen Dank! :)

#8 - 2017-05-16 16:14 - hidden

- Description updated

- Status changed from Customer Feedback Required to To Be Closed

@Florian, Ticket kann geschlossen werden.

#9 - 2017-05-16 16:14 - hidden

- File deleted (image001.gif)

#10 - 2017-05-17 12:18 - hidden

- Subject changed from ADTFS-46726 FW: CAN-FD to ADTFS-46726 Provide CAN-FD structure

- Status changed from To Be Closed to Closed

#11 - 2018-01-25 12:10 - hidden

- Project changed from 7 to Public Support

- Private changed from Yes to No

Files

canfd_types.h	14.4 KB	2017-05-11	hidden