CANAPI library

Version
6/7/2010 12:25:00 PM

Table of Contents

Data Structure Index

Data Structures

Here are the data structures with brief descriptions:

File Index

File List

Here is a list of all documented files with brief descriptions:

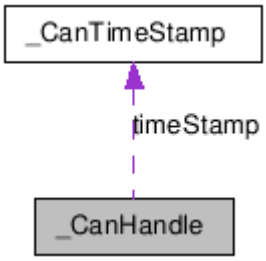
Data Structure Documentation

_CanHandle Struct Reference

Definition of CAN handle.

`#include <canapi.h>`

Collaboration diagram for _CanHandle:



Data Fields

- UINT32 majorVersionCanH
  Major version of "can.h" included by client. NOT IN USE.
- CanTimeStamp timeStamp

Timestamp for last read message.

- int nCan
  Can channel (n from CANn) NOT IN USE.
- int socket
  Socket value for socketCAN usage.

Detailed Description

Definition of CAN handle.

Definition at line 50 of file canapi.h.

The documentation for this struct was generated from the following file:

- canapi.h

_CanMsg Struct Reference

Definition of CAN message.

`#include <canapi.h>`

Data Fields

- CanMsgId id
  Id of can message.
- CanFrameType frameType
  Frame type of CAN message.
- UINT8 length
  Length of CAN message data.
- UINT8 data [CAN_MAX_MSG_LENGTH]
  Data of Can message.

Detailed Description

Definition of CAN message.

Definition at line 66 of file canapi.h.

The documentation for this struct was generated from the following file:

- canapi.h

_CanTimeStamp Struct Reference

Definition of CAN Timestamp.

`#include <canapi.h>`

Data Fields

- UINT32 low
  USeconds.
- UINT32 high
  Seconds.

Detailed Description

Definition of CAN Timestamp.

Definition at line 43 of file canapi.h.

The documentation for this struct was generated from the following file:

- canapi.h

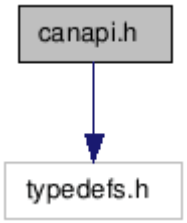
File Documentation

canapi.h File Reference

CANAPI definitions.

`#include "typedefs.h"`

Include dependency graph for canapi.h:



Data Structures

- struct _CanTimeStamp
  Definition of CAN Timestamp.
- struct _CanHandle
  Definition of CAN handle.
- struct _CanMsg
  Definition of CAN message.

Typedefs

- typedef struct _CanTimeStamp CanTimeStamp
  Definition of CAN Timestamp.
- typedef struct _CanHandle * CanHandle
  Definition of CAN handle.
- typedef UINT32 CanMsgId
  Definition of CAN message id.
- typedef UINT32 CanFrameType
  Definition of CAN frame type.
- typedef struct _CanMsg CanMsg
  Definition of CAN message.
- typedef UINT32 CanDrvBaudrate
  Definition of CAN driver baudrate (bit/s).

Enumerations

- enum ECCCanBaudrate { CCCAN_BAUDRATE_1M = 1000000, CCCAN_BAUDRATE_800K = 800000, CCCAN_BAUDRATE_500K = 500000, CCCAN_BAUDRATE_250K = 250000, CCCAN_BAUDRATE_125K = 125000, CCCAN_BAUDRATE_100K = 100000, CCCAN_BAUDRATE_50K = 50000, CCCAN_BAUDRATE_20K = 20000, CCCAN_BAUDRATE_AUTO = 0 }
  Definition of CAN Baudrates.
- enum ECanError { CAN_ERROR_NO_ERROR = 0, CAN_ERROR_INIT_FAILED = -1, CAN_ERROR_INVALID_CAN_INTERFACE = -2, CAN_ERROR_GLOBAL_STD_MASK_SET_FAILED = -3, CAN_ERROR_GLOBAL_EXT_MASK_SET_FAILED = -4, CAN_ERROR_RX_OBJ_SET_FAILED = -5, CAN_ERROR_INVALID_BAUDRATE = -6, CAN_ERROR_AUTO_BAUD_FAILED = -7, CAN_ERROR_BAUDRATE_NOT_SET = -8, CAN_ERROR_AUTO_BAUD_ACTIVE = -9, CAN_ERROR_NO_MSG_RECEIVED = -10, CAN_ERROR_INVALID_HANDLE = -11, CAN_ERROR_DEVICE_NOT_OPEN = -12, CAN_ERROR_SEND_TIMED_OUT = -13, CAN_ERROR_MSG_SEND_FAILED = -14, CAN_ERROR_RECEIVE_TIMED_OUT = -15, CAN_ERROR_NOT_SUPPORTED_FEATURE = -255 }
  Definition of CAN error return values.

Functions

- CanHandle CanOpen (LPCTSTR pNetName)
  a function for opening CAN-interface.
- BOOL CanClose (CanHandle hInterface)
  a function for closing CAN-interface.
- BOOL CanSend (CanHandle hInterface, CanMsg *pCanMsg, DWORD dataLength, BOOL bRtr)
  a function for sending CAN message
- BOOL CanReceive (CanHandle hInterface, CanMsg *pCanMsg, LPDWORD pDataLength, CanMsgId *pCanMsgSel, DWORD milliseconds)
  a function for sending CAN message
- BOOL CanSetBaudrate (CanHandle hInterface, ECCCanBaudrate eSpeed)
  a function for setting CAN controller baudrate
- BOOL CanGetBaudrate (CanHandle hInterface, ECCCanBaudrate *eSpeed)
  a function for getting CAN controller baudrate
- ECanError CanGetLastError (CanHandle hInterface)
  a function for getting last CAN error

- BOOL CanGetLastTimeStamp (CanHandle hInterface, CanTimeStamp *pTimeStamp)
  a function for getting timestamp of last received message

Detailed Description

CANAPI definitions.

Author:
Teemu Keskinarkaus / CC Systems Oy

Definition in file canapi.h.

Enumeration Type Documentation

enum ECanError

Definition of CAN error return values.

Enumerator:

CAN_ERROR_NO_ERROR No error.
CAN_ERROR_INIT_FAILED Initialization failed.
CAN_ERROR_INVALID_CAN_INTERFACE Given CAN-interface is invalid.
CAN_ERROR_GLOBAL_STD_MASK_SET_FAILED Global Standard mask set failed.
CAN_ERROR_GLOBAL_EXT_MASK_SET_FAILED Global Extended mask set failed.
CAN_ERROR_RX_OBJ_SET_FAILED Receive Object set failed.
CAN_ERROR_INVALID_BAUDRATE Given baudrate is invalid.
CAN_ERROR_AUTO_BAUD_FAILED Baudrate set failed.
CAN_ERROR_BAUDRATE_NOT_SET Baudrate is not set.
CAN_ERROR_AUTO_BAUD_ACTIVE Auto baudrate is active.
CAN_ERROR_NO_MSG_RECEIVED No message received.
CAN_ERROR_INVALID_HANDLE Given handler is invalid.
CAN_ERROR_DEVICE_NOT_OPEN Device is not open.
CAN_ERROR_SEND_TIMED_OUT Send timed out.
CAN_ERROR_MSG_SEND_FAILED Sending message failed.
CAN_ERROR_RECEIVE_TIMED_OUT Receive timed out.
CAN_ERROR_NOT_SUPPORTED_FEATURE Feature is not supported.

Definition at line 78 of file canapi.h.

enum ECCCanBaudrate

Definition of CAN Baudrates.

Enumerator:

CCCAN_BAUDRATE_1M 1Mbit

CCCAN_BAUDRATE_800K 800kbit

CCCAN_BAUDRATE_500K 500kbit

CCCAN_BAUDRATE_250K 250kbit

CCCAN_BAUDRATE_125K 125kbit

CCCAN_BAUDRATE_100K 100kbit

CCCAN_BAUDRATE_50K 50kbit

CCCAN_BAUDRATE_20K 20kbit

CCCAN_BAUDRATE_AUTO AUTO.

Definition at line 21 of file canapi.h.

Function Documentation

BOOL CanClose (CanHandle *hInterface*)

a function for closing CAN-interface.

Parameters:

hInterface Handle for CAN-interface

Returns:

TRUE if operation succeeded otherwise FALSE. If operation failed then "GetLastError" can be used to get the information of the error.

BOOL CanGetBaudrate (CanHandle *hInterface*, ECCCanBaudrate * *eSpeed*)

a function for getting CAN controller baudrate

Parameters:

hInterface Handle for CAN-interface
eSpeed Pointer to Baudrate variable

Returns:

TRUE if operation succeeded otherwise FALSE. If operation failed then "GetLastError" can be used to get the information of the error.

ECanError CanGetLastError (CanHandle *hInterface*)

a function for getting last CAN error

Parameters:

hInterface Handle for CAN-interface

Returns:

ECanError error code

BOOL CanGetLastTimeStamp (CanHandle *hInterface*, CanTimeStamp * *pTimeStamp*)

a function for getting timestamp of last received message

Parameters:

hInterface Handle for CAN-interface
pTimeStamp Pointer tothe timestamp

Returns:

TRUE if operation succeeded otherwise FALSE. If operation failed then "GetLastError" can be used to get the information of the error.

CanHandle CanOpen (LPCTSTR *pNetName*)

a function for opening CAN-interface.

This function must be called before using any other CAN-interface functions.

Parameters:

pNetName Name of the interface to open

Returns:

A handle to opened interface or NULL if operation failed If operation failed then "GetLastError" can be used to get the information of the error.

BOOL CanReceive (CanHandle *hInterface*, CanMsg * *pCanMsg*, LPDWORD *pDataLength*, CanMsgId * *pCanMsgSel*, DWORD *milliseconds*)

a function for sending CAN message

Parameters:

hInterface Handle for CAN-interface
pCanMsg A pointer to the received message.
pDataLength A pointer to the number of data bytes
pCanMsgSel A pointer to an array specifying a selection of which messages that should be received. The first element, (pCanMsgSel[0]), should specify the number of CAN message IDs in the array. If pCanMsgSel[0] is positive then any of IDs in the array will be received. If pCanMsgSel[0] is negative then any of IDs that is not in the array will be received. If any message is reqested then NULL should be supplied.
milliseconds Timeout interval. The function returns if the interval elapses, even if no messages are received. If milliseconds is zero, the function checks if there are any messages and returns immediately. If milliseconds is INFINITE, the function does not return until a message is received.

Returns:

TRUE if operation succeeded otherwise FALSE. If operation failed then "GetLastError" can be used to get the information of the error.

BOOL CanSend (CanHandle *hInterface*, CanMsg * *pCanMsg*, DWORD *dataLength*, BOOL *bRtr*)

a function for sending CAN message

Parameters:

hInterface Handle for CAN-interface
pCanMsg Pointer to CAN Message to be send
dataLength Data length in CAN message
bRtr Should the message be send as Remote Frame

Returns:

TRUE if operation succeeded otherwise FALSE. If operation failed then "GetLastError" can be used to get the information of the error.

BOOL CanSetBaudrate (CanHandle *hInterface*, ECCCanBaudrate *eSpeed*)

a function for setting CAN controller baudrate

Parameters:

hInterface Handle for CAN-interface
eSpeed Baudrate value

Returns:

TRUE if operation succeeded otherwise FALSE. If operation failed then "GetLastError" can be used to get the information of the error. In case of AutoBaud, will return FALSE until speed is found. It can be read using CanGetBaudrate - function.

Index