

27/10/2018

CRUSB_Spartan_1_3_0.dll (Function list)

History:

| | |
|--------------------------|--|
| CRUSB_Spartan_V1_3_0.dll | Add BRPE for CAN bit rate calculation. |
| CRUSB_Spartan_V1_2_0.dll | Fix Tx set on echo message. |
| CRUSB_Spartan_V1_1_0.dll | New names of function and parameters in functions and results. |
| CRUSB_Spartan_V1_0_0.dll | first working library |

Functions:

| | |
|-----------------|---|
| CRUSBOpen | Open USB communication with CRUSB device |
| CRUSBClose | Close USB communication with CRUSB device |
| CRUSBInfo | Information about CRUSB |
| DLLInfo | Info about DLL |
| CAN_Init | Init CAN communication |
| CAN_Read | Read one CAN frame from Rx buffer |
| CAN_Write | Write one CAN frame to Tx buffer |
| CAN_Control | CAN communication control |
| CAN_Status | Status of CAN |
| CAN_Status_Ctrl | CAN Status ON or OFF |
| CAN_Rx_Clear | Clear Rx Buffer in DLL FIFO |
| CAN_Tx_Clear | Clear Tx Buffer in DLL FIFO |

Using these functions, it is possible to make communication on CAN bus for 11bit and 29bit frames.

Service

| DELPHI | | Borlad C++ |
|--|--------------------------------|---|
| DLLInfo(DLL_INFO:DLL_INFO_):smallint; | | short DLLInfo(DLL_INFO* DLL_INFO_) |
| Type | Function | |
| Parameters | DLL_INFO: pointer to structure | unsigned char |
| | DLL_INFO.Version :widerstring; | WideString |
| Result | 0 | ok |
| Description | Return version of DLL. | |

USB Communication

| DELPHI | | Borlad C++ |
|--|---|--|
| CRUSBOpen(Device:Byte):smallint | | short CRUSBOpen(unsigned char Device) |
| Type | Function | |
| Parameters | Device:Byte | unsigned char |
| | 0 | Device number to connect (only one device) |
| Result | 0 255 | connected disconnected |
| Description | Calling this function is necessary to establish the connection via USB. | |

| DELPHI | | Borlad C++ |
|---|---|--|
| CRUSBClose(Device:Byte):smallint | | short CRUSBClose(unsigned char Device) |
| Type | Function | |
| Parameters | Device: Byte | unsigned char |
| | 0 | Device number to disconnect (only value 0 allowed) |
| Result | 0 | ok |
| Description | Calling this function is necessary to close USB communication with CRUSB. | |

| DELPHI | | Borlad C++ |
|---|--|---|
| CRUSBInfo(CRUSB_info:pointer):smallint | | short CRUSBInfo(CRUSB_INFO* CRUSB_INFO_) |
| Type | Procedure | |
| Parameters | CRUSB_info: pointer to structure | |
| | CRUSB_info.Name: widerstring; | 'CRUSB Spartan' |
| | CRUSB_info.SN: widerstring; | 'Hardware: x.xx' |
| | CRUSB_info.HW: widerstring; | 'Software: x.xx' |
| | CRUSB_info.SW: widerstring; | 'S/N:xxxxxxx' |
| | CRUSB_info.DLL: widerstring; | 'DLL vx.x.x' |
| Result | 0 | ok |
| Description | Basic information about CRUSB (static info from device). | |

CAN Communication

| DELPHI | | Borlad C++ |
|--|--|--|
| CAN_Init(BTR0_BTR1:word;BRPE:byte ;Par:word;CAN_BitRate_Calc:dword):smallint | | short CAN_Init(short BTR0_BTR1, unsigned char BRPE,unsigned char Par, ui32) |
| Type | Functions | |
| Parameters | BTR0_BTR1: Word (CAN boudrate) | short |
| | 0x1C0A 0x1C1D 0x1C0E 0x1C0B 0x1C05 0x1C02 0x1B01 0x0502 | PAR=0x18 20kb/s 50kb/s 100kb/s 125kb/s 250kb/s 500kb/s 800kb/s 1000kb/s |
| | BRPE:Byte (BRPE – 0 - 15) 1 0 0 0 0 0 0 0 0 | Unsigned char 20kb/s 50kb/s 100kb/s 125kb/s 250kb/s 500kb/s 800kb/s 1000kb/s |
| | Par: BYTE | unsigned char |
| | 0x0C 0x18 | 12 MHz 24 MHz |
| | CAN_BitRate_Calc:dword decimal value of CAN bitrate used for busload calculation | Uint 32 |
| Result | 0 255 | connected disconnected |
| Description | This function connects to CAN bus, Par - parameter of CAN controller frquency (Do not use random values in BTR0_BTR1 because it can makes unexpected operation in CRUSB) For other values than listed above try to calculate according BTR0 and BTR1 for C_CAN Bosch) http://www.bittiming.can-wiki.info/#C_CAN | |

| DELPHI | | Borlad C++ |
|--|--|---|
| CAN_Control(mode:byte):smallint | | int CAN_Control(unsigned char mode) |
| Type | Function | |
| Parameters | 0 1 2 3 | CAN Stop CAN Start CAN Start echo CAN Listen |
| Result | 0 | ok |
| Description | This function disconnect from CAN bus. | |

| DELPHI | | Borlad C++ | | | | | | | | | | | | | | | | | |
|----------------------------------|--|--|--|---|---|---|---|---|---|---|---|------|-----|-------|-----|---|---|---|---|
| CAN_Read(CAN_MSG:PCAN_MSG):int32 | | int CAN_Read(CAN_MSG_RX* CAN_MSG_RX_) | | | | | | | | | | | | | | | | | |
| Type | Function | | | | | | | | | | | | | | | | | | |
| Parameters | CAN_RxMSG - Pointer to structure | | | | | | | | | | | | | | | | | | |
| | CAN_RxMSG.DATE_YY:BYTE; CAN_RxMSG.DATE_MM:BYTE; CAN_RxMSG.DATE_DD:BYTE; CAN_RxMSG.TIME_HH:BYTE; CAN_RxMSG.TIME_MM:BYTE; CAN_RxMSG.TIME_SS:BYTE; CAN_RxMSG.TIME_us:WORD; CAN_RxMSG.ID_DW:DWORD; CAN_RxMSG.DLC:byte; CAN_RxMSG.DATA:array [0..7] of byte; CAN_RxMSG.RTR:boolean; CAN_RxMSG.EXT:boolean; CAN_RxMSG.RX_TX:boolean; CAN_RxMSG.MSG_TYPE:byte; | unsigned char unsigned char unsigned char unsigned char unsigned char unsigned int unsigned long unsigned char unsigned char[8] bool bool bool unsigned char | Frame ID Number of bytes in CAN frame Data <table border="1" data-bbox="1045 560 1420 649"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>ECHO</td><td>RTR</td><td>29bit</td><td>DIR</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> 0 – Rx , 1 - TX | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | ECHO | RTR | 29bit | DIR | 0 | 0 | 0 | 0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | | | | | |
| ECHO | RTR | 29bit | DIR | 0 | 0 | 0 | 0 | | | | | | | | | | | | |
| Result | lint32 | int | number of frames left in buffer (max 100 000 frames) | | | | | | | | | | | | | | | | |
| Description | Every call this function returns one CAN frame in structure CAN_MSG, and as a result the number of CAN frames left in buffer. CAN_MSG.RX_TX is set when Echo ON (see CAN_Control). | | | | | | | | | | | | | | | | | | |

| DELPHI | | Borlad C++ | | | | | | | | | | | | | | | | | |
|--------------------------------------|---|---|---|---|---|---|---|---|---|---|---|------|-----|-------|-----|---|---|---|---|
| CAN_Write(CAN_MSG:TCAN_MSG):smallint | | short CAN_Write(CAN_MSG_TX _CAN_MSG_TX) | | | | | | | | | | | | | | | | | |
| Type | Function | | | | | | | | | | | | | | | | | | |
| Parameters | CAN_TxMSG - structure | | | | | | | | | | | | | | | | | | |
| | CAN_TxID_DW:dword CAN_TxDLC:byte CAN_TxDATA:array [0..7] of byte CAN_TxMSG_TYPE:byte | unsigned short unsigned char unsigned char unsigned char | CAN frame ID DLC CAN DATA Type of frame <table border="1" data-bbox="1029 1489 1404 1579"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>ECHO</td><td>RTR</td><td>29bit</td><td>DIR</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | ECHO | RTR | 29bit | DIR | 0 | 0 | 0 | 0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | | | | | |
| ECHO | RTR | 29bit | DIR | 0 | 0 | 0 | 0 | | | | | | | | | | | | |
| Result | 0 | ok | | | | | | | | | | | | | | | | | |
| Description | Every call send one can frame to Tx buffer. | | | | | | | | | | | | | | | | | | |

| DELPHI | | Borlad C++ | |
|---|---|---------------------------------------|---------------------------|
| CAN_Status(CAN_INFO:PCAN_INFO):smallint | | short CAN_Status(CAN_STAT* CAN_STAT_) | |
| Type | Function | | |
| Parameters | CAN_INFO - pointer to structure | | |
| | CAN_INFO.w_Boudrate: Word; | unsigned short | CAN baudrate |
| | CAN_INFO.w_Busload: Word; | unsigned short | Current CAN bus load |
| | CAN_INFO.b_Listen: Boolean; | bool | Listen mode |
| | CAN_INFO.b_Rx_Error: Boolean; | bool | Rx Error appear |
| | CAN_INFO.b_Tx_Error: Boolean; | bool | Tx Error appear |
| | CAN_INFO.b_BusOFF: Boolean; | bool | CAN Bus OFF appear |
| | CAN_INFO.b_Passive: Boolean; | bool | CAN Bus paasive appear |
| | CAN_INFO.w_RX_Error_cnt: Word; | unsigned short | Number of Rx error frames |
| | CAN_INFO.w_TX_Error_cnt: Word; | unsigned short | Number of Tx error frames |
| | CAN_INFO.B_LEC: Byte; | unsigned char | LEC error ¹ |
| | CAN_INFO.B_Buffer_Load: Byte; | unsigned char | Not implemented |
| Result | 0 | ok | |
| Description | Returns the information about CAN controller. | | |

| DELPHI | | Borlad C++ | |
|-------------------------------------|---|---|------------|
| CAN_Status_Ctrl(mode:byte):smallint | | short CAN_Status_Ctrl(unsigned char mode) | |
| Type | Function | | |
| Parameters | mode | | |
| | 0 | | Status OFF |
| | 1 | | Status ON |
| Result | 0 | ok | |
| Description | Every call send one can frame to Tx buffer. | | |

| DELPHI | | Borlad C++ | |
|-------------------------|--|--------------------------|--|
| CAN_Rx_Clear():smallint | | short CAN_Rx_Clear(void) | |
| Type | Function | | |
| Parameters | - | | |
| Result | 0 | ok | |
| Description | Function clears DLL Rx CAN messages FIFO | | |

| DELPHI | | Borlad C++ | |
|-------------------------|--|--------------------------|--|
| CAN_Tx_Clear():smallint | | short CAN_Tx_Clear(void) | |
| Type | Function | | |
| Parameters | - | | |
| Result | 0 | ok | |
| Description | Function clears DLL Tx CAN messages FIFO | | |