USB CAN Bus DII Instructions

Revision: V1.00 Date: February 08, 2024

www.holtek.com

Table of Contents

1. HTCAN_OpenCom 3
2. HTCAN_CloseCom 3
3. HTCAN_OpenCanDevice 3
4. HTCAN_CloseCanDevice 3
5. HTCAN_SetCanBitRate 4
6. HTCAN_TansCanFrame 4
7. HTCAN_TansRTRCanFrame 5
8. HTCAN_SetAcceptanceMask 5
9. HTCAN_SetAcceptanceCode 5
10. HTCAN_GetStatusFlag 6
11. HTCAN_GetFirmwareVersion 6
12. SetGetTimeFun 6
13. SetPassiveReadFunc 6
14. GetErrCode 6

This DLL file contains a total of 14 functions, which are described as follows.

1. HTCAN_OpenCom

This function is used to open a serial device.

```
BOOL HTCAN_OpenCom(UINT nComIndex);
```

Parameter

nComIndex	Serial port index; e.g., to index COM13, this parameter value is 13

Return Value

TRUE	Succeeded to open the serial port
FALSE	Failed to open the serial port

2. HTCAN_CloseCom

This function is used to close a serial device.

```
BOOL HTCAN_CloseCom();
```

Parameter

None

Return Value

Fixed at 1, meaningless

3. HTCAN_OpenCanDevice

This function is used to open a CAN Bus device and an O/l/L command will be sent, which corresponds to Normal mode, Loopback mode or Listen Only mode respectively.

```
BOOL HTCAN_OpenCanDevice(BYTE byOpenMode);
```

Parameter

byOpenMode	Mode parameter: MODE_NORMAL, MODE_LOOPBACK, MODE_LISTEN

Return Value

TRUE	Succeeded to open the device
FALSE	Failed to open the device

4. HTCAN_CloseCanDevice

This function is used to close the CAN Bus device and a C command will be sent.

```
BOOL HTCAN_CloseCanDevice()
```

Parameter

None

Return Value

TRUE	Succeeded to close the device
FALSE	Failed to close the device

5. HTCAN_SetCanBitRate

This function is used to set the CAN Bus baud rate and an S command will be sent.

```
BOOL HTCAN_SetCanBitRate(BYTE byRate);
```

Parameter

byRate Baud rate index

The following values are available:

```
#define BR_5K     0
#define BR_10K    1
#define BR_20K    2
#define BR_50K    3
#define BR_100K   4
#define BR_125K   5
#define BR_250K   6
#define BR_500K   7
#define BR_800K   8
#define BR_1M     9
```

Return Value

TRUE	Succeeded to set the baud rate
FALSE	Failed to set the baud rate

6. HTCAN_TansCanFrame

This function is used to transmit a standard/extended CAN frame through the CAN bus, and a t/T command will be sent.

```
BOOL HTCAN_TansCanFrame(BYTE byMode, DWORD dwId, BYTE *pbyData, BYTE byDataSize,
BYTE *pbyLoopBackData, BYTE &byLoopBackDataSize);
```

Parameter

byMode	Two modes to select: TRANS_STD, TRANS_EXT
dwId	CAN frame identifier, max. 0x7FF for TRANS_STD mode, max. 0x1FFFFFFF for TRANS_EXT mode
pbyData	Data to be transmitted
byDataSize	Length of the transmitted data, up to 8 bytes, the exceeded data are ignored
pbyLoopBackData	Stores the returned data if there is return data from the device
byLoopBackDataSize	Stores the number of returned bytes if there is return data from the device

Return Value

TRUE	Succeeded
FALSE	Failed

7. HTCAN_TansRTRCanFrame

This function is used to transmit a standard/extended remote request through the CAN bus, and an r/R command will be sent.

```
BOOL HTCAN_TansRTRCanFrame(BYTE byMode, DWORD dwId, BYTE byDataSize, BYTE
*pbyReadData, BYTE &byReadSize)
```

Parameter

byMode	Two modes to select: TRANS_STD, TRANS_EXT
dwId	Same to “HTCAN_TansCanFrame”
byDataSize	Number of bytes to be read
pbyReadData	The data read
byReadSize	The actual length of the content read

Return Value

TRUE	Succeeded
FALSE	Failed

8. HTCAN_SetAcceptanceMask

This function is used to set acceptance filter mask and an m command will be sent.

```
BOOL HTCAN_SetAcceptanceMask(BYTE byMode, DWORD dwMask)
```

Parameter

byMode	Two modes to select: TRANS_STD, TRANS_EXT
dwMask	CAN mask, max. 0x7FF for TRANS_STD, max. 0x1FFFFFFFFF for TRANS_EXT

Return Value

TRUE	Succeeded
FALSE	Failed

9. HTCAN_SetAcceptanceCode

This function is used to set acceptance filter code and an M command will be sent.

```
BOOL HTCAN_SetAcceptanceCode(BYTE byMode, DWORD dwCode);
```

Parameter

byMode	Two modes to select: TRANS_STD, TRANS_EXT
dwCode	CAN code; max. 0x7FF for TRANS_STD, max. 0x1FFFFFFFFF for TRANS_EXT

Return Value

TRUE	Succeeded
FALSE	Failed

10. HTCAN_GetStatusFlag

This function is used to obtain the CAN Bus status and an F command will be sent.

```
BOOL HTCAN_GetStatusFlag(BYTE &byFlag);
```

Parameter

byFlag	CAN bus status

Return Value

TRUE	Succeeded
FALSE	Failed

11. HTCAN_GetFirmwareVersion

This function is used to obtain the current firmware version, and a v command will be sent.

```
BOOL HTCAN_GetFirmwareVersion (DWORD &infoValue);
```

Parameter

infoValue	Stores the returned version number

Return Value

TRUE	Succeeded
FALSE	Failed

12. SetGetTimeFun

This function is used to obtain the time of sending command and the time of receiving data.

```
void SetGetTimeFun(GetTimeFunc pFunc);
```

Parameter

pFunc	Its prototype: typedef void (*GetTimeFunc)(DWORD dwTime,BOOL bRead);

13. SetPassiveReadFunc

This function is used to obtain other serial port information except Ack and Data required for commands.

```
void SetPassiveReadFunc(GetPassiveReadFunc pFunc);
```

Parameter

pFunc	Its prototype: typedef void (*GetPassiveReadFunc)(BYTE *cBackData, DWORD &dwReadSize);

14. GetErrCode

This function is used to obtain the function execution error code.

```
DWORD GetErrCode();
```

Return Value

The return values are as follows:

0xFF	Failed to execute “CreateFile”
0xFE	The specified serial port is invalid
0x01	The characters sent do not match the actual number of characters to be sent
0x02	Failed to execute “GetOverlappedResult”

0x03	Timed out
0x04	Failed to execute “ClearCommError”
0x05	Serial port data reception stopped
0x06	The number of characters returned is incorrect
0x07	The returned string format does not match the one set in the protocol
Others	Error status value returned by GetLastError