

Date	Auth	Notice
2023/04/06	Y. OGAWA	1st release

目次

目次	1
注意点	2
ファイル構成	2
関数	3
GetInterfaceCount	3
SetInterface	3
SetBaudrate	4

注意点

- ・ 本 DLL は、CANabh3.dll と入れ替え利用が可能な、別インターフェース（WacoGiken 社製）用となります。
- ・ 本書では CANabh3.dll との相違点のみ説明されています。基本的な情報は、CANabh3.dll 側のプロジェクト付属文書を御参照願います。

ファイル構成

本 DLL は、CANabh3 と以下の相違点があります。
元資料となる、CANabh3 側の資料も合わせてご確認ください。

ファイル名	内容
IxxatV2.cpp IxxatV2.h	本 DLL では存在しません。 このファイルは、HMS 社の USB-to-CAN V2 デバイスを制御する為のファイルです。
WacoCanUsb.cpp WacoCanUsb.h	このファイルは、WacoGiken 社の CAN インターフェースを制御する為のファイルです。
Crc.cpp Crc.h	CCITT-CRC を計算する為のクラスです。 送信用パケット構築、又は受信パケットの検証に必要となります。

インターフェースの利用準備

本 DLL では以下のインターフェースのみに対応しています。

メーカー	WacoGiken
メーカーURL	https://www.wacogiken.co.jp/
名称	CAN インターフェース
準備	インターフェースの入手方法は、要問い合わせとなります。 PC に接続すれば、COM ポート扱いで認識します。
本書更新時の デバイスドライバ	不用

関数

基本的に関数説明は、CANabh3.dll の説明書を御確認下さい。
本 DLL 専用の関数は有りませんが、一部仕様が異なる関数のみ説明が有ります。

GetInterfaceCount

概要	使用可能な CAN インターフェース数を取得
詳細	InitInstance で指定した「使用したいインターフェース」に対して、現時点で利用可能な本数(PC に接続されているデバイス数)を取得します。
構文	CANABH3API int32_t GetInterfaceCount()
パラメータ	無し
戻り値	PC の「COM ポート数」が戻ります。
注意点等	

SetInterface

概要	CAN 回線に接続するインターフェースを指定						
詳細	CAN 回線に接続するインターフェースを指定。						
構文	CANABH3API int32_t SetInterface(int32_t nDeviceNum)						
パラメータ	<table><tr><th>変数名</th><th>内容</th></tr><tr><td>nDeviceNum</td><td>開く対象の COM ポート番号を指定して下さい。 ケーブルを接続した時、COM3 になるケーブルを選択する場合は、3 を指定します。</td></tr></table>	変数名	内容	nDeviceNum	開く対象の COM ポート番号を指定して下さい。 ケーブルを接続した時、COM3 になるケーブルを選択する場合は、3 を指定します。		
変数名	内容						
nDeviceNum	開く対象の COM ポート番号を指定して下さい。 ケーブルを接続した時、COM3 になるケーブルを選択する場合は、3 を指定します。						
戻り値	<table><tr><th>戻り値</th><th>内容</th></tr><tr><td>0</td><td>正常終了</td></tr><tr><td>上記以外</td><td>異常終了</td></tr></table>	戻り値	内容	0	正常終了	上記以外	異常終了
戻り値	内容						
0	正常終了						
上記以外	異常終了						
注意点等	複数本接続した場合、使われていない COM ポート番号が自動的に割り振られます。 確認するには、Windows の機能であるデバイスマネージャを御利用下さい。						

SetBaudrate

概要	CAN 回線で使用する通信速度を指定																												
詳細	CAN 回線で使用する通信速度を指定します。																												
構文	CANABH3API void SetBaudrate(uint32_t nBaudrateKbps)																												
パラメータ	<table><tr><th>変数名</th><th colspan="2">内容</th></tr><tr><td rowspan="12">nBaudrateKbps</td><td colspan="2">通信速度 [Kbps] を以下から指定しますが、実際の通信速度は、CAN インターフェース側の物理的なスイッチで設定する必要が有ります。</td></tr><tr><th>値</th><th>通信速度</th></tr><tr><td>10</td><td>10 [Kbps]</td></tr><tr><td>20</td><td>20 [Kbps]</td></tr><tr><td>50</td><td>50 [Kbps]</td></tr><tr><td>100</td><td>100 [Kbps]</td></tr><tr><td>125</td><td>125 [Kbps]</td></tr><tr><td>250</td><td>250 [Kbps]</td></tr><tr><td>500</td><td>500 [Kbps]</td></tr><tr><td>800</td><td>800 [Kbps]</td></tr><tr><td>1000</td><td>1000 [Kbps]</td></tr></table>			変数名	内容		nBaudrateKbps	通信速度 [Kbps] を以下から指定しますが、実際の通信速度は、CAN インターフェース側の物理的なスイッチで設定する必要が有ります。		値	通信速度	10	10 [Kbps]	20	20 [Kbps]	50	50 [Kbps]	100	100 [Kbps]	125	125 [Kbps]	250	250 [Kbps]	500	500 [Kbps]	800	800 [Kbps]	1000	1000 [Kbps]
変数名	内容																												
nBaudrateKbps	通信速度 [Kbps] を以下から指定しますが、実際の通信速度は、CAN インターフェース側の物理的なスイッチで設定する必要が有ります。																												
	値	通信速度																											
	10	10 [Kbps]																											
	20	20 [Kbps]																											
	50	50 [Kbps]																											
	100	100 [Kbps]																											
	125	125 [Kbps]																											
	250	250 [Kbps]																											
	500	500 [Kbps]																											
	800	800 [Kbps]																											
	1000	1000 [Kbps]																											
	戻り値	無し																											
注意点等	CAN 回線に接続する前に必ず設定する必要が有りますが、実際の通信速度に影響されません。実際の通信速度は、CAN インターフェース側に搭載されているスイッチで指定が必要です。																												

CAN インターフェースのプロトコル

構成

名称	開始コード	フラグ	ID	データ	CRC	終端コード
内容	STX	フラグ	CAN-ID	データ	CCITT-CRC	ETX
長さ [bytes]	1	2	4	0 - 8	2	1

解説

項目名	STX
内容	開始文字となるコード バイナリで 02h となる

項目名	フラグ
内容	以下の値を 16 進数 2 桁の ANSI 文字列で扱う。 bit7 - bit5 : 常に 0 bit4 : 拡張 ID フラグ (0..標準 ID 1..拡張 ID) bit3 - bit0 : データ長
例	拡張 ID として 8 バイトのデータを送信する場合 bit7 - bit5 : 000 bit4 : 1 bit3 - bit0 : 1000 合わせると 2 進数で 00011000 となり、これを 16 進数 2 桁の文字列に変換 最終的な文字は、18 となる

項目名	CAN-ID
内容	CAN-ID を 16 進数 8 桁の ANSI 文字列で扱う。
例	CAN-ID が、0xEF0102 (発信元が 02、送信先が 01、コードが EFh) の場合 最終的な文字列は、00EF0102 となる

項目名	データ
内容	データ部分を 16 進数 2 桁単位の ANSI 文字列で扱う。 最小はデータ無し (0 文字)、最大は 8 バイト (16 文字となる)

項目名	CCITT-CRC
内容	フラグ (2 文字目) からデータ末尾迄を 1 バイト単位のバイナリデータとして扱い、CCITT-CRC で算出して 16 進数 4 桁の文字列として扱う。 CCITT-CRC 仕様は以下の通り。 長さ (16bit)、コード (0x1021)、値 (反転しない)、方向 (左回り) C++ で利用する場合は、本プロジェクトに含まれる CRC.cpp/h の利用を推奨。
例	フラグ ("18")、CAN-ID ("00EF0102")、データ ("C309C30900000000") の場合、 算出結果は 0xCC4B となり、最終的な文字列は CC4B となる。

項目名	ETX
内容	終端文字となるコード バイナリで 03h となる