

## 関数リファレンス for CANabh3.dll

| Date       | Auth     | Notice                    |
|------------|----------|---------------------------|
| 2023/03/09 | Y. OGAWA | 1st release               |
| 2023/03/29 | Y. OGAWA | 2nd release<br>誤記訂正、補足追加等 |
|            |          |                           |
|            |          |                           |

## 目次

|                        |    |
|------------------------|----|
| 目次                     | 1  |
| 注意点                    | 3  |
| 環境                     | 4  |
| ファイル構成                 | 5  |
| 基本的な利用方法               | 7  |
| CANAbh3dll.cpp/h の利用方法 | 8  |
| 列挙子                    | 9  |
| 構造体                    | 10 |
| 関数                     | 12 |
| InitInstance           | 12 |
| ExitInstance           | 13 |
| GetInterfaceCount      | 14 |
| SetInterface           | 15 |
| GetCurrentInterface    | 16 |
| SetOpenTimeout         | 17 |
| SetSendTimeout         | 18 |
| SetRecvTimeout         | 19 |
| SetHostID              | 20 |
| GetHostID              | 21 |
| SetBaudrate            | 22 |
| GetBaudrate            | 23 |
| GetTm                  | 24 |
| GetCounter             | 25 |
| GetCANError            | 26 |
| ResetCANError          | 27 |
| abh3_can_init          | 28 |
| abh3_can_port_init     | 29 |
| abh3_cmd_init          | 30 |
| abh3_can_cmdAY         | 31 |
| abh3_can_cmdBX         | 31 |
| abh3_can_cmd           | 32 |
| abh3_can_inSet         | 33 |
| abh3_can_inBitSet      | 34 |
| abh3_can_cmdAndopSet   | 35 |
| abh3_can_reqBRD        | 36 |
| abh3_can_trans         | 37 |
| abh3_can_copylastdata  | 38 |
| abh3_can_resetlastdata | 39 |
| abh3_can_read          | 40 |
| abh3_can_flush         | 41 |
| abh3_can_finish        | 42 |
| cnvVel2CAN             | 43 |
| cnvCAN2Vel             | 44 |
| cnvCur2CAN             | 45 |
| cnvCAN2Cur             | 46 |
| cnvCAN2Load            | 47 |
| cnvCAN2Analog          | 48 |
| cnvCAN2Volt            | 49 |
| 値の単位                   | 50 |
| 非対称通信について              | 51 |

|           |    |
|-----------|----|
| 实装例 ..... | 52 |
|-----------|----|

## 注意点

- ・ 本 DLL プロジェクトはソースコードを含んだ Visual Studio 用のプロジェクトとして提供されます。  
利用する Visual Studio は、バージョン 2019 を想定しています。
- ・ 本 DLL の利用には、以下の知識がある事が前提となります。  
標準 DLL を利用する為の知識
- ・ 本 DLL から HMS 製の USB-to-CAN V2 インターフェースのみが利用可能です。  
HMS 社から USB-to-CAN V2 インターフェースのデバイスドライバを取得し、本 DLL を利用する  
アプリケーションの開発環境と実行環境へ、それぞれインストールする必要があります
- ・ 本 DLL は、32bit アプリケーション用の DLL として設計されていますが、  
実行環境は一般向け Windows の x64 バージョン (WOW64 搭載の Windows10, 11) を想定しています。
- ・ 本 DLL は、前身である CANa31.dll に非対称通信を行う為の機能を追加して改造した物になります。

Visual Studio 及び Windows は Microsoft の商標です。

## 環境

本 DLL の作成環境と想定利用環境は以下の通りです

| 要素      | 作成環境   | 想定利用環境   |
|---------|--|--|
| OS      | Windows10 pro 64bit<br>version 21H2                  | Windows10 pro/home 64bit<br>version 21H2 又はそれ以降                    |
| CPU     | Intel i9-10900X                                      | Intel 系 CPU(*1)  |
| コンパイラ   | Microsoft Visual Studio 2019 pro<br>version 16.11.15 | Microsoft Visual Studio 2019                                       |
| DLL 利用先 |  | 32bit アプリケーション<br>・ MFC アプリケーション(32bit)<br>・ Win32 アプリケーション(32bit) |

(\*1) Intel 第 12 世代（又はそれ以降）の CPU 利用時、動作に問題が有る場合は、  
実行環境の UEFI-BIOS 設定で、CPU の E コアを無効化する事で改善する可能性があります。  
詳細は、実行環境の PC マニュアルを参照願います。

## ファイル構成

本 DLL は以下のファイルで構成されます

| ファイル名  | 内容   |
|--|--|
| ABH3.cpp<br>ABH3.h   | ABH3 ドライバ固有機能 C++クラス   |
| Can1939.h  | CAN J1939 仕様のデータ作成用 C++クラス   |
| CANabh3.cpp<br>CANabh3.h                                     | 本 DLL でエクスポートされる関数が格納された C ソースコード<br>ユーザーが利用する関数は、このコード内の関数が出入口となります   |
| CANabh3.def  | 本 DLL でエクスポートされる関数を定義したファイル<br>本 DLL では、このファイルをプロジェクト設定で明示的に指定してあります   |
| CANabh3.rc   | DLL に内包されるリソース定義<br>バージョン情報等が含まれます   |
| CANabh3dll.cpp<br>CANabh3dll.h                               | 本 DLL をユーザー側でダイナミックロードする場合に、関数を楽に扱う為のコード。<br>ユーザー側の上位アプリケーションを MFC で作成する場合に利用可能。<br>本 DLL 構築時にこのソースは利用されません。 |
| CanIF.cpp<br>CanIF.h   | 本 DLL で扱うインターフェースの制御を行う C++クラス<br>利用可能な CAN インターフェースは、本クラスから継承して実装します  |
| dllmain.cpp  | DLL エントリ<br>アタッチ／デタッチ時に処理が必要な場合は、本ソースコードを変更して御利用下さい  |
| IxxatV2.cpp<br>IxxatV2.h                                     | HMS 社製、USB-to-CAN v2 インターフェースの制御クラス  |
| typedef.h  | 本 DLL でユーザーが使用する列挙子・構造体の定義   |
| packfloat.cpp<br>packfloat.h                                 | 実数系に packfloat（当社作成の特殊変数）を利用する場合に必要となるコード。   |
| resource.h<br>pch.cpp<br>pch.h<br>targetvar.h<br>framework.h | Visual Studio 利用時に自動作成されるファイル。<br>一部変更済み。  |
| readme.md  | Git サーバ用表題ファイル（履歴記載有り）   |
| CANabh3.sln<br>CANabh3.vcproj                                | Visual Studio 用のソリューション/プロジェクトファイル。<br>本プロジェクトは、Visual Studio 2019 で構築しています。                                 |

## 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               | HMS  |
| メーカーURL            | <a href="https://www.ixxat.com/ja/">https://www.ixxat.com/ja/</a>  |
| 名称                 | USB-to-CAN V2  |
| 準備                 | HMS 社のサイトから、Windows 用のファイル(zip)をダウンロードします<br>解凍後に出てくる実行ファイルを、開発環境及び実行環境で実行し、<br>上記名称用のデバイスドライバ（開発環境なら SDK を追加）を選択してインストール。   |
| 本書更新時の<br>デバイスドライバ | ダウンロード URL<br><a href="https://www.ixxat.com/ja/technical-support/support/windows-driver-software">https://www.ixxat.com/ja/technical-support/support/windows-driver-software</a><br><br>バージョン<br>4.0.1003.0 |

## 基本的な利用方法

本 DLL の利用想定アプリケーションと利用方法は、以下となります

| No. | アプリケーション種類     | 利用方法  |
|-----|----------------|---|
| 1   | Win32 アプリケーション | <p>本プロジェクトの生成物（DLL/LIB）をユーザー側のアプリケーションでスタティック又はダイナミックリンクで御利用下さい。</p> <p>但し、Visual Studio 2019 以外のコンパイラを利用される場合は、本 DLL プロジェクトをお客様の環境で再ビルドしてから御利用下さい。</p>   |
| 2   | MFC アプリケーション   | <p>本プロジェクトの生成物（DLL/LIB）をユーザー側のアプリケーションでスタティック又はダイナミックリンクで御利用下さい。</p> <p>但し、Visual Studio 2019 以外のコンパイラを利用される場合は、本 DLL プロジェクトをお客様の環境で再ビルドしてから御利用下さい。</p> <p>DLL の動的ロードを行う場合、プロジェクト内の CanABH3dll.cpp/h を利用する事で、本 DLL を楽に扱う事が可能です。<br/>（「CANabh3dll.cpp/h の利用方法」の項を参照）</p> |



## CANabh3dll.cpp/h の利用方法

プロジェクトにソースのコピーを追加し、利用可能にしてください。

### コード例

```
#include "CANabh3dll.h"
static CAN_FUNCLIST g_func;

#define CABLE (0)
#define TARGETID (1)
#define HOSTID (2)
#define BAUDRATE (250)
#define TM_OPEN (1000)
#define TM_SEND (100)
#define TM_RECV (100)
#define A_SRVON (3)

int foo()
{
    //受信用構造体
    CANABH3_RESULT result;

    //DLLの読み込み
    HANDLE hDLL = LoadLibrary("CANabh3.dll");
    if (hDLL == NULL)
        return (-1); //DLL読み込みエラー

    //DLLに含まれる関数の取得 (CANabh3dll.cpp内に関数実体有り)
    GetFunctions(hDLL, &g_func);

    //DLL関数の使用例
    g_func.InitInstance(0); //初期化関数の呼び出し

    //インターフェースの指定 (何本目のケーブル)
    g_func.SetInterface(CABLE);

    //ホスト番号設定
    g_func.SetHostID(HOSTID);

    //ボーレート設定値[Kbps]取得し設定
    g_func.SetBaudrate(BAUDRATE);

    //タイムアウト系を設定
    g_func.SetOpenTimeout(TM_OPEN);
    g_func.SetSendTimeout(TM_SEND);
    g_func.SetRecvTimeout(TM_RECV);

    //インターフェースを開いて指令を初期化、受信有り
    g_func.abh3_can_init(TARGETID, &result);

    //デバイス番号1の機器に対してA軸サーボONと指令有効、受信有り
    g_func.abh3_can_cmd(TARGETID, A_SRVON, 0, &result);

    //指令値500.0を機器側の速度値に変換
    int16_t nValue = g_func.cnvVel2CAN(500.0);

    //デバイス番号1の機器に対して、指令値をA軸に送信、受信有り
    g_func.abh3_can_cmdAY(TARGETID, nValue, &result);

    //時間待ち(5秒)
    Sleep(5000);

    //デバイス番号1の機器に対してA軸サーボOFFと指令無効、受信有り
    g_func.abh3_can_cmd(TARGETID, 0, 0, &result);

    //CAN回線を切断
    g_func.abh3_can_finish();

    //DLL解放する為の呼び出し
    g_func.ExitInstance();

    //DLL開放
    FreeLibrary(hDLL);

    return(0);
}
```

## 列挙子

```
//パケット種類の指定用
typedef enum
{
    UNKNOWN_PACKET    = 0,    //内部用（ユーザーでは利用しない）
    SINGLE_PACKET      = 1,    //シングルパケット
    BROADCAST_PACKET   = 2,    //ブロードキャストパケット
    ANY_PACKET         = 0,    //パケット種類を問わない
} PACKETTYPE;
```

## 構造体

//受信に使用する構造体（受信情報+受信データ8バイト）

```
typedef struct _CANABH3_RESULT
{
    //受信データのCAN-ID
    uint32_t nID; //CANID
    union
    {
        //内部用
        uint8_t raw[8];
        //受信要素
        //シングルパケット(DP0)
        struct _DPOR
        {
            int16_t nBackAY; //A/Y帰還
            int16_t nBackBX; //B/X帰還
            uint32_t nCtrlBit; //制御フラグ
        } DPOR;
        //ブロードキャストパケット(0)
        struct _BR0
        {
            uint32_t nErrorBit; //異常フラグ
            uint32_t nWarnBit; //警告フラグ
        } BR0;
        //ブロードキャストパケット(1)
        struct _BR1
        {
            uint32_t nIoBit; //IOフラグ
            uint32_t nCtrlBit; //入力フラグ
        } BR1;
        //ブロードキャストパケット(2)
        struct _BR2
        {
            int16_t nOrderSpeedAY; //A/Y速度指令
            int16_t nOrderSpeedBX; //B/X速度指令
            int16_t nBackSpeedAY; //A/Y速度帰還
            int16_t nBackSpeedBX; //B/X速度帰還
        } BR2;
        //ブロードキャストパケット(3)
        struct _BR3
        {
            int16_t nOrderCurrentAY; //A/Y電流指令
            int16_t nOrderCurrentBX; //B/X電流指令
            int16_t nLoadA; //A負荷率
            int16_t nLoadB; //B負荷率
        } BR3;
        //ブロードキャストパケット(4)
        struct _BR4
        {
            int32_t nCountPulseA; //Aパルス積算値
            int32_t nCountPulseB; //Bパルス積算値
        } BR4;
        //ブロードキャストパケット(5)
        struct _BR5
        {
            int16_t nAnalog0; //アナログ入力0
            int16_t nAnalog1; //アナログ入力1
            int16_t nPowerMain; //主電源電圧
            int16_t nPowerCtrl; //制御電源電圧
        } BR5;
        //ブロードキャストパケット(6)
        struct _BR6
        {
            float nMonitor0; //モニタ0データ
            float nMonitor1; //モニタ1データ
        } BR6;
    } u;
} CANABH3_RESULT, *pCANABH3_RESULT;
```

```

//最終受信データ用構造体
typedef struct _CANABH3_LASTRECV
{
    //シングルパケット (DP0)
    struct _DP0R
    {
        int16_t    nBackAY;           //A/Y帰還
        int16_t    nBackBX;           //B/X帰還
        uint32_t    nCtrlBit;         //制御フラグ
    } DP0R;

    //ブロードキャストパケット (0)
    struct _BR0
    {
        uint32_t    nErrorBit;        //異常フラグ
        uint32_t    nWarnBit;        //警告フラグ
    } BR0;

    //ブロードキャストパケット (1)
    struct _BR1
    {
        uint32_t    nIoBit;           //IOフラグ
        uint32_t    nCtrlBit;        //入力フラグ
    } BR1;

    //ブロードキャストパケット (2)
    struct _BR2
    {
        int16_t    nOrderSpeedAY;     //A/Y速度指令
        int16_t    nOrderSpeedBX;     //B/X速度指令
        int16_t    nBackSpeedAY;      //A/Y速度帰還
        int16_t    nBackSpeedBX;      //B/X速度帰還
    } BR2;

    //ブロードキャストパケット (3)
    struct _BR3
    {
        int16_t    nOrderCurrentAY;   //A/Y電流指令
        int16_t    nOrderCurrentBX;   //B/X電流指令
        int16_t    nLoadA;            //A負荷率
        int16_t    nLoadB;            //B負荷率
    } BR3;

    //ブロードキャストパケット (4)
    struct _BR4
    {
        int32_t    nCountPulseA;      //Aパルス積算値
        int32_t    nCountPulseB;      //Bパルス積算値
    } BR4;

    //ブロードキャストパケット (5)
    struct _BR5
    {
        int16_t    nAnalog0;          //アナログ入力0
        int16_t    nAnalog1;          //アナログ入力1
        int16_t    nPowerMain;        //主電源電圧
        int16_t    nPowerCtrl;        //制御電源電圧
    } BR5;

    //ブロードキャストパケット (6)
    struct _BR6
    {
        float      nMonitor0;         //モニタ0データ
        float      nMonitor1;         //モニタ1データ
    } BR6;

    //各格納場所の更新フラグ
    struct _UPDATE
    {
        uint8_t    nUpdate;           //0以外で値が更新されている (受信している)
    } update[8];                     //0..DP0R 1..BR0 2..BR1 .... 7..BR6
} CANABH3_LASTRECV, *pCANABH3_LASTRECV;

//PACK_FLOAT用構造体
typedef struct _PACK_FLOAT
{
    union
    {
        struct _PACK_FLOAT_INFO
        {
            //LSB側から記載 (処理系に注意)
            int32_t kasuu:21;          // 仮数部 (-999999~0~999999)
            int32_t shousuutenn:4;     // 少数点 (-8~0~7)
            int32_t shisuu:7;         // 指数部 (-64~0~63)
        } info;
        int32_t    nDirectData;        // 直接アクセス用
    } u;
} *pPACK_FLOAT, PACK_FLOAT;

```

## 関数

### InitInstance

|        |   |                     |  |     |    |  |        |                      |  |   |          |   |                     |
|--------|---|---------------------|--|-----|----|--|--------|----------------------|--|---|----------|---|---------------------|
| 概要     | インターフェースの利用開始   |                     |  |     |    |  |        |                      |  |   |          |   |                     |
| 詳細     | インターフェースの利用を開始します。  |                     |  |     |    |  |        |                      |  |   |          |   |                     |
| 構文     | CANABH3API void InitInstance(int32_t nIFnum)  |                     |  |     |    |  |        |                      |  |   |          |   |                     |
| パラメータ  | <table><tr><td>変数名</td><td colspan="2">内容</td></tr><tr><td rowspan="3">nIFnum</td><td colspan="2">使用するインターフェースを指定して下さい</td></tr><tr><td>値</td><td>インターフェース</td></tr><tr><td>0</td><td>Ixxat USB-to-CAN V2</td></tr></table> |                     |  | 変数名 | 内容 |  | nIFnum | 使用するインターフェースを指定して下さい |  | 値 | インターフェース | 0 | Ixxat USB-to-CAN V2 |
| 変数名    | 内容  |                     |  |     |    |  |        |                      |  |   |          |   |                     |
| nIFnum | 使用するインターフェースを指定して下さい  |                     |  |     |    |  |        |                      |  |   |          |   |                     |
|        | 値   | インターフェース            |  |     |    |  |        |                      |  |   |          |   |                     |
|        | 0   | Ixxat USB-to-CAN V2 |  |     |    |  |        |                      |  |   |          |   |                     |
| 戻り値    | 無し  |                     |  |     |    |  |        |                      |  |   |          |   |                     |
| 注意点等   | この時点では、CAN 回線にはまだ接続されていません。<br>本 DLL でサポートされるインターフェースは、Ixxat USB-to-CAN V2 のみとなります。   |                     |  |     |    |  |        |                      |  |   |          |   |                     |

## ExitInstance

|       |   |
|-------|---|
| 概要    | インターフェースの利用終了   |
| 詳細    | インターフェースの利用を終了し、DLL を開放可能な状態にします。                                 |
| 構文    | CANABH3API void ExitInstance()                                    |
| パラメータ | 無し  |
| 戻り値   | 無し  |
| 注意点等  | CAN 回線を切断してなかった場合、自動的に切断されます<br>既にインターフェースの利用を終了している場合は、何も処理しません。 |

## 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 に接続されている「本 DLL で利用可能な CAN インターフェース数」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>開く対象のインターフェース番号として、<br/>何本目のインターフェース（0 開始）を指定します</td></tr></table>           |  | 変数名 | 内容 | nDeviceNum | 開く対象のインターフェース番号として、<br>何本目のインターフェース（0 開始）を指定します |      |      |
| 変数名        | 内容   |  |     |    |            |   |      |      |
| nDeviceNum | 開く対象のインターフェース番号として、<br>何本目のインターフェース（0 開始）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       | <p>複数本接続した場合、接続した順に番号が割り振られます。<br/>但し、PC にインターフェースを接続した状態で PC を起動した場合、又は再起動した場合等、常に同じ接続順になるとは限らない事に注意が必要です。</p> <p>基本的には 1 本のみ利用する事を推奨します。</p> |  |     |    |            |   |      |      |



## GetCurrentInterface

|       |  |  |
|-------|--|--|
| 概要    | CAN 回線に接続したインターフェース番号を取得                 |  |
| 詳細    | CAN 回線に接続した時点で指定されていたインターフェース番号を取得。      |  |
| 構文    | CANABH3API int32_t GetCurrentInterface() |  |
| パラメータ | 無し                                       |  |
| 戻り値   |  |  |
|       | 戻り値                                      | 内容                                     |
|       | 0 未満                                     | CAN 回線に接続していない                         |
|       | 上記以外                                     | CAN 回線に接続した時点で、SetInterface 関数に指定していた値 |
| 注意点等  |  |  |

## SetOpenTimeout

SetOpenTimeout

| 概要         | CAN 回線の接続タイムアウト時間を設定   |  |     |    |            |                                     |
|------------|--|--|-----|----|------------|-------------------------------------|
| 詳細         | CAN 回線に接続する時、接続要求から接続完了迄の許容時間を指定します。   |  |     |    |            |                                     |
| 構文         | CANABH3API void SetOpenTimeout(uint32_t nTimeoutMS)  |  |     |    |            |                                     |
| パラメータ      | <table><tr><th>変数名</th><th>内容</th></tr><tr><td>nTimeoutMS</td><td>CAN 回線へ接続する処理の許容時間[ms]<br/>推奨値は 1000</td></tr></table> |  | 変数名 | 内容 | nTimeoutMS | CAN 回線へ接続する処理の許容時間[ms]<br>推奨値は 1000 |
| 変数名        | 内容   |  |     |    |            |                                     |
| nTimeoutMS | CAN 回線へ接続する処理の許容時間[ms]<br>推奨値は 1000  |  |     |    |            |                                     |
| 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。  |  |     |    |            |                                     |

## SetSendTimeout

| 概要         | CAN 回線の送信タイムアウト時間を設定  |  |     |    |            |                                    |
|------------|---|--|-----|----|------------|------------------------------------|
| 詳細         | CAN 回線へデータ送信する時、送信要求から送信完了迄の許容時間を指定します。   |  |     |    |            |                                    |
| 構文         | CANABH3API void SetSendTimeout(uint32_t nTimeoutMS)   |  |     |    |            |                                    |
| パラメータ      | <table><tr><th>変数名</th><th>内容</th></tr><tr><td>nTimeoutMS</td><td>CAN 回線へのデータ送信許容時間[ms]<br/>推奨値は 1000</td></tr></table> |  | 変数名 | 内容 | nTimeoutMS | CAN 回線へのデータ送信許容時間[ms]<br>推奨値は 1000 |
| 変数名        | 内容  |  |     |    |            |                                    |
| nTimeoutMS | CAN 回線へのデータ送信許容時間[ms]<br>推奨値は 1000  |  |     |    |            |                                    |
| 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。   |  |     |    |            |                                    |

## SetRecvTimeout

| 概要         | CAN 回線の受信タイムアウト時間を設定  |  |     |    |            |                                    |
|------------|---|--|-----|----|------------|------------------------------------|
| 詳細         | CAN 回線からデータ受信する時、受信要求から受信完了迄の許容時間を指定します。  |  |     |    |            |                                    |
| 構文         | CANABH3API void SetRecvTimeout(uint32_t nTimeoutMS)   |  |     |    |            |                                    |
| パラメータ      | <table><tr><th>変数名</th><th>内容</th></tr><tr><td>nTimeoutMS</td><td>CAN 回線からの受信処理許容時間[ms]<br/>推奨値は 1000</td></tr></table> |  | 変数名 | 内容 | nTimeoutMS | CAN 回線からの受信処理許容時間[ms]<br>推奨値は 1000 |
| 変数名        | 内容  |  |     |    |            |                                    |
| nTimeoutMS | CAN 回線からの受信処理許容時間[ms]<br>推奨値は 1000  |  |     |    |            |                                    |
| 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。   |  |     |    |            |                                    |

## SetHostID

| 概要    | CAN 回線上の通信ホストアドレスを設定   |  |     |    |       |  |
|-------|--|--|-----|----|-------|--|
| 詳細    | PC が使用する通信アドレスを指定します。  |  |     |    |       |  |
| 構文    | CANABH3API void SetHostID(uint8_t nAdrs)   |  |     |    |       |  |
| パラメータ | <table><tr><th>変数名</th><th>内容</th></tr><tr><td>nAdrs</td><td>通信ホスト (PC) のアドレスを指定します<br/>通信ホストから送信する場合は、他の機器と重複する値を指定しないで下さい、又 00h と FFh は利用不可です。</td></tr></table> |  | 変数名 | 内容 | nAdrs | 通信ホスト (PC) のアドレスを指定します<br>通信ホストから送信する場合は、他の機器と重複する値を指定しないで下さい、又 00h と FFh は利用不可です。 |
| 変数名   | 内容   |  |     |    |       |  |
| nAdrs | 通信ホスト (PC) のアドレスを指定します<br>通信ホストから送信する場合は、他の機器と重複する値を指定しないで下さい、又 00h と FFh は利用不可です。   |  |     |    |       |  |
| 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。<br>他の機器の通信を傍受する為だけに利用する場合は、傍受対象の機器 ID を指定して下さい。  |  |     |    |       |  |

## GetHostID

| 概要        | CAN 回線上の通信ホストアドレスを取得   |  |     |    |           |                                      |
|-----------|--|--|-----|----|-----------|--------------------------------------|
| 詳細        | PC が使用する通信アドレスを取得します。  |  |     |    |           |                                      |
| 構文        | CANABH3API uint8_t GetHostID()   |  |     |    |           |                                      |
| パラメータ     | 無し   |  |     |    |           |                                      |
| 戻り値       | <table><tr><th>戻り値</th><th>内容</th></tr><tr><td>01h - FEh</td><td>SetHostID 関数で指定された通信ホストアドレスが<br/>戻ります</td></tr></table> |  | 戻り値 | 内容 | 01h - FEh | SetHostID 関数で指定された通信ホストアドレスが<br>戻ります |
| 戻り値       | 内容   |  |     |    |           |                                      |
| 01h - FEh | SetHostID 関数で指定された通信ホストアドレスが<br>戻ります   |  |     |    |           |                                      |
| 注意点等      |  |  |     |    |           |                                      |

## 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</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><tr><td></td><td></td><td></td></tr></table> |            |  | 変数名 | 内容 |  | nBaudrateKbps | 通信速度[Kbps]を以下から指定します |  | 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  |            |  |     |    |  |               |                      |  |   |      |    |          |    |          |    |          |     |           |     |           |     |           |     |           |     |           |      |            |  |  |  |
|               | 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          | <p>CAN 回線に接続する前に必ず設定する必要が有ります。</p> <p>パラメータで指定可能な値以外は、通信速度に設定出来ません。<br/>ある程度高速な通信速度を指定した場合、ケーブル及び終端抵抗の品質によっては正しく通信出来ない場合が有ります。</p> <p>CAN 回線接続後に通信速度を変更する場合は、一度 CAN 回線を切断(abh3_can_finish)してから、本関数で新しい速度を設定し、再度 CAN 回線に接続(abh3_can_port_open)して下さい。</p>   |            |  |     |    |  |               |                      |  |   |      |    |          |    |          |    |          |     |           |     |           |     |           |     |           |     |           |      |            |  |  |  |

## GetBaudrate

|       |  |
|-------|--|
| 概要    | 設定した通信速度を取得  |
| 詳細    | 設定済みの通信速度を取得します。   |
| 構文    | CANABH3API uint32_t GetBaudrate()  |
| パラメータ | 無し   |
| 戻り値   | SetBaudrate で指定された通信速度[Kbps]が戻ります。   |
| 注意点等  | CAN 回線接続中に SetBaudrate で通信速度を設定しても、実際の通信速度は変わりませんが、本関数の戻り値は新しく設定した値が戻るので注意が必要です。 |



## GetTm

|       |   |
|-------|---|
| 概要    | 時間を[ms]単位で取得  |
| 詳細    | PC が起動した時間を 0 として、現在迄の時間を[ms]単位で取得します。  |
| 構文    | CANABH3API uint32_t GetTm()   |
| パラメータ | 無し  |
| 戻り値   | PC が起動してから現在迄の時間が[ms]単位で戻ります。   |
| 注意点等  | <p>32bit が最大の為、49.7 日程度でオーバーフローして 0 に戻ります。<br/>         本関数は、連続稼働前提のシステムで利用する事は、非推奨です。</p> <p>非力な PC では、時間取得自体に処理時間が掛かる場合が有り、本関数を利用する事で、アプリケーションのパフォーマンスに影響する場合があります。</p> |

## GetCounter

|       |   |
|-------|---|
| 概要    | 送受信データ量をビット単位で取得  |
| 詳細    | 最後に本関数を利用してから、次に利用する迄に送受信を行った CAN 通信で、CAN 回線上で送受信したデータ量（ヘッダ含む）をビット単位で取得します。                                   |
| 構文    | CANABH3API uint32_t GetCounter ()   |
| パラメータ | 無し  |
| 戻り値   | 送受信量がビット単位で戻ります   |
| 注意点等  | 本関数は、上位アプリケーションから CAN 回線の使用帯域率を求める為に用意されましたが、厳密な値では無い事に注意が必要です。（誤差が有ります）<br>（計算に利用するには、精度の良い周期でアクセスする必要が有ります） |

## GetCANError

| 概要    | インターフェースにエラーが発生しているか調べます  |  |     |    |   |       |   |                       |      |             |
|-------|---|--|-----|----|---|-------|---|-----------------------|------|-------------|
| 詳細    | インターフェースにエラーが発生しているか調べます  |  |     |    |   |       |   |                       |      |             |
| 構文    | CANABH3API uint32_t GetCANError()   |  |     |    |   |       |   |                       |      |             |
| パラメータ | 無し  |  |     |    |   |       |   |                       |      |             |
| 戻り値   | <table><tr><th>戻り値</th><th>内容</th></tr><tr><td>0</td><td>エラー無し</td></tr><tr><td>2</td><td>インターフェースのアクセス自体に問題がある</td></tr><tr><td>上記以外</td><td>その他のエラーが発生中</td></tr></table> |  | 戻り値 | 内容 | 0 | エラー無し | 2 | インターフェースのアクセス自体に問題がある | 上記以外 | その他のエラーが発生中 |
| 戻り値   | 内容  |  |     |    |   |       |   |                       |      |             |
| 0     | エラー無し   |  |     |    |   |       |   |                       |      |             |
| 2     | インターフェースのアクセス自体に問題がある   |  |     |    |   |       |   |                       |      |             |
| 上記以外  | その他のエラーが発生中   |  |     |    |   |       |   |                       |      |             |
| 注意点等  | 以下の条件でインターフェースにエラーが発生します。<br>・ CAN バスに PC（ホスト）以外が無い状態で、送信を続ける。<br>・ CAN バスの信号劣化（ケーブル品質に対してボーレートが高い）   |  |     |    |   |       |   |                       |      |             |

## ResetCANError

| 概要    | インターフェースに発生しているエラーの解除処理を実行します  |  |     |    |   |  |   |                       |      |          |
|-------|--|--|-----|----|---|--|---|-----------------------|------|----------|
| 詳細    | インターフェースに発生しているエラーの解除処理を実行します<br>この関数でエラーが解除出来る保証は有りません。<br>可能であれば、一度インターフェースを閉じて、再度開き直す処理での対応を推奨します。  |  |     |    |   |  |   |                       |      |          |
| 構文    | CANABH3API uint32_t ResetCANerror ()   |  |     |    |   |  |   |                       |      |          |
| パラメータ | 無し   |  |     |    |   |  |   |                       |      |          |
| 戻り値   | <table><tr><th>戻り値</th><th>内容</th></tr><tr><td>0</td><td>エラー解除処理の実行は正常終了。<br/>但し、エラーが解除出来たかどうかは不明。</td></tr><tr><td>2</td><td>インターフェースのアクセス自体に問題がある</td></tr><tr><td>上記以外</td><td>エラー解除不可。</td></tr></table> |  | 戻り値 | 内容 | 0 | エラー解除処理の実行は正常終了。<br>但し、エラーが解除出来たかどうかは不明。 | 2 | インターフェースのアクセス自体に問題がある | 上記以外 | エラー解除不可。 |
| 戻り値   | 内容   |  |     |    |   |  |   |                       |      |          |
| 0     | エラー解除処理の実行は正常終了。<br>但し、エラーが解除出来たかどうかは不明。   |  |     |    |   |  |   |                       |      |          |
| 2     | インターフェースのアクセス自体に問題がある  |  |     |    |   |  |   |                       |      |          |
| 上記以外  | エラー解除不可。   |  |     |    |   |  |   |                       |      |          |
| 注意点等  | Ixxat USB-to-CAN V2 ケーブルでは多くの場合、インターフェースを一度閉じないとエラー解除出来ない場合があります。  |  |     |    |   |  |   |                       |      |          |

## abh3\_can\_init

| 概要         | 指令の初期化及び CAN 回線へ接続  |     |     |           |             |        |   |            |   |
|------------|---|-----|-----|-----------|-------------|--------|---|------------|---|
| 詳細         | <p>以下の順で処理が行われます。</p> <p>(1) 以下の要素を一括設定</p> <table> <tr> <th>要素名</th><th>設定値</th></tr> <tr> <td>A/Y 指令</td><td>0</td></tr> <tr> <td>B/X 指令</td><td>0</td></tr> <tr> <td>入力(bit 対応)</td><td>0</td></tr> </table> <p>(2) SetInterface で指定されたインターフェースを利用して、CAN 回線を接続</p> <p>(3) シングルパケットとして、上記要素を格納したパケットを送信</p> <p>(4) (返信の格納領域が指定有りなら) 返答が有る物として受信</p> | 要素名 | 設定値 | A/Y 指令    | 0           | B/X 指令 | 0   | 入力(bit 対応) | 0 |
| 要素名        | 設定値   |     |     |           |             |        |   |            |   |
| A/Y 指令     | 0   |     |     |           |             |        |   |            |   |
| B/X 指令     | 0   |     |     |           |             |        |   |            |   |
| 入力(bit 対応) | 0   |     |     |           |             |        |   |            |   |
| 構文         | CANABH3API int32_t abh3_can_init(uint8_t nTargetID, pCANABH3_RESULT pPtr)   |     |     |           |             |        |   |            |   |
| パラメータ      | <table> <tr> <th>変数名</th><th>内容</th></tr> <tr> <td>nTargetID</td><td>通信先の ID を指定</td></tr> <tr> <td>pPtr</td><td>送信に対する返信の格納領域<br/>NULL 指定時は送信のみ行います<br/>受信時は、pPtr-&gt;u.DPOR に格納されます</td></tr> </table>  | 変数名 | 内容  | nTargetID | 通信先の ID を指定 | pPtr   | 送信に対する返信の格納領域<br>NULL 指定時は送信のみ行います<br>受信時は、pPtr->u.DPOR に格納されます |            |   |
| 変数名        | 内容  |     |     |           |             |        |   |            |   |
| nTargetID  | 通信先の ID を指定   |     |     |           |             |        |   |            |   |
| pPtr       | 送信に対する返信の格納領域<br>NULL 指定時は送信のみ行います<br>受信時は、pPtr->u.DPOR に格納されます   |     |     |           |             |        |   |            |   |
| 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>0 以外</td><td>異常終了時のエラーコード</td></tr> </table>   | 戻り値 | 内容  | 0         | 正常終了        | 0 以外   | 異常終了時のエラーコード  |            |   |
| 戻り値        | 内容  |     |     |           |             |        |   |            |   |
| 0          | 正常終了  |     |     |           |             |        |   |            |   |
| 0 以外       | 異常終了時のエラーコード  |     |     |           |             |        |   |            |   |
| 注意点等       |   |     |     |           |             |        |   |            |   |

## abh3\_can\_port\_init

| 概要    | CAN 回線に接続  |  |     |    |   |      |      |              |
|-------|--|--|-----|----|---|------|------|--------------|
| 詳細    | CAN 回線に接続。   |  |     |    |   |      |      |              |
| 構文    | CANABH3API int32_t abh3_can_port_init(void)  |  |     |    |   |      |      |              |
| パラメータ | 無し   |  |     |    |   |      |      |              |
| 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>0 以外</td><td>異常終了時のエラーコード</td></tr></table> |  | 戻り値 | 内容 | 0 | 正常終了 | 0 以外 | 異常終了時のエラーコード |
| 戻り値   | 内容   |  |     |    |   |      |      |              |
| 0     | 正常終了   |  |     |    |   |      |      |              |
| 0 以外  | 異常終了時のエラーコード   |  |     |    |   |      |      |              |
| 注意点等  | SetInterface で指定されたインターフェースを利用して、CAN 回線に接続します。<br>CAN 回線に接続するのみで、何も送信しません。<br>指令等も初期化されない事に注意が必要です。                        |  |     |    |   |      |      |              |

## abh3\_cmd\_init

| 概要         | 指令の初期化と送信   |     |     |           |             |        |   |            |   |
|------------|---|-----|-----|-----------|-------------|--------|---|------------|---|
| 詳細         | <p>以下の順で処理が行われます。</p> <p>(1) 以下の要素を一括設定</p> <table> <tr> <th>要素名</th><th>設定値</th></tr> <tr> <td>A/Y 指令</td><td>0</td></tr> <tr> <td>B/X 指令</td><td>0</td></tr> <tr> <td>入力(bit 対応)</td><td>0</td></tr> </table> <p>(2) シングルパケットとして、上記要素を格納したパケットを送信</p> <p>(3) (返信の格納領域が指定有りなら) 返答が有る物として受信</p> | 要素名 | 設定値 | A/Y 指令    | 0           | B/X 指令 | 0   | 入力(bit 対応) | 0 |
| 要素名        | 設定値   |     |     |           |             |        |   |            |   |
| A/Y 指令     | 0   |     |     |           |             |        |   |            |   |
| B/X 指令     | 0   |     |     |           |             |        |   |            |   |
| 入力(bit 対応) | 0   |     |     |           |             |        |   |            |   |
| 構文         | CANABH3API int32_t abh3_cmd_init(uint8_t nTargetID, pCANABH3_RESULT pPtr)   |     |     |           |             |        |   |            |   |
| パラメータ      | <table> <tr> <th>変数名</th><th>内容</th></tr> <tr> <td>nTargetID</td><td>通信先の ID を指定</td></tr> <tr> <td>pPtr</td><td>送信に対する返信の格納領域<br/>NULL 指定時は送信のみ行います<br/>受信時は、pPtr-&gt;u.DPOR に格納されます</td></tr> </table>  | 変数名 | 内容  | nTargetID | 通信先の ID を指定 | pPtr   | 送信に対する返信の格納領域<br>NULL 指定時は送信のみ行います<br>受信時は、pPtr->u.DPOR に格納されます |            |   |
| 変数名        | 内容  |     |     |           |             |        |   |            |   |
| nTargetID  | 通信先の ID を指定   |     |     |           |             |        |   |            |   |
| pPtr       | 送信に対する返信の格納領域<br>NULL 指定時は送信のみ行います<br>受信時は、pPtr->u.DPOR に格納されます   |     |     |           |             |        |   |            |   |
| 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>0 以外</td><td>異常終了時のエラーコード</td></tr> </table>   | 戻り値 | 内容  | 0         | 正常終了        | 0 以外   | 異常終了時のエラーコード  |            |   |
| 戻り値        | 内容  |     |     |           |             |        |   |            |   |
| 0          | 正常終了  |     |     |           |             |        |   |            |   |
| 0 以外       | 異常終了時のエラーコード  |     |     |           |             |        |   |            |   |
| 注意点等       | 本関数は、既に CAN 回線へ接続している状態での利用が前提となります。  |     |     |           |             |        |   |            |   |

abh3\_can\_cmdAY

abh3\_can\_cmdBX

| 概要        | 指令の送信（軸別）  |  |      |      |           |            |      |                |      |   |
|-----------|--|--|------|------|-----------|------------|------|----------------|------|---|
| 詳細        | 指定を送信します。  |  |      |      |           |            |      |                |      |   |
| 構文        | CANABH3API int32_t abh3_can_cmdAY(uint8_t nTargetID, int16_t cmd, pCANABH3_RESULT pPtr)<br>CANABH3API int32_t abh3_can_cmdBX(uint8_t nTargetID, int16_t cmd, pCANABH3_RESULT pPtr)   |  |      |      |           |            |      |                |      |   |
| パラメータ     | <table><tr><th>変数名</th><th>内容</th></tr><tr><td>nTargetID</td><td>送信先 ID</td></tr><tr><td>cmd</td><td>A/Y 又は B/X 指令値</td></tr><tr><td>pPtr</td><td>送信に対する返信の格納領域<br/>NULL を指定すると送信のみで戻ります<br/>受信時は、pPtr-&gt;u. DPOR に格納されます</td></tr></table>                                       |  | 変数名  | 内容   | nTargetID | 送信先 ID     | cmd  | A/Y 又は B/X 指令値 | pPtr | 送信に対する返信の格納領域<br>NULL を指定すると送信のみで戻ります<br>受信時は、pPtr->u. DPOR に格納されます |
| 変数名       | 内容   |  |      |      |           |            |      |                |      |   |
| nTargetID | 送信先 ID   |  |      |      |           |            |      |                |      |   |
| cmd       | A/Y 又は B/X 指令値   |  |      |      |           |            |      |                |      |   |
| pPtr      | 送信に対する返信の格納領域<br>NULL を指定すると送信のみで戻ります<br>受信時は、pPtr->u. DPOR に格納されます  |  |      |      |           |            |      |                |      |   |
| 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>0 以外</td><td>異常終了時のエラーコード</td></tr></table>   |  | 戻り値  | 内容   | 0         | 正常終了       | 0 以外 | 異常終了時のエラーコード   |      |   |
| 戻り値       | 内容   |  |      |      |           |            |      |                |      |   |
| 0         | 正常終了   |  |      |      |           |            |      |                |      |   |
| 0 以外      | 異常終了時のエラーコード   |  |      |      |           |            |      |                |      |   |
| 注意点等      | <p>A/Y 指令送信では過去の B/X 指令値と一緒に送信されます。<br/>B/X 指令送信では過去の A/Y 指令値と一緒に送信されます。<br/>指令として与える値は、適切な変換関数を使用して下さい。<br/>又、制御対象がその指令単位になっている事を先に確認して下さい。</p> <table><tr><th>指令単位</th><th>変換関数</th></tr><tr><td>速度</td><td>cnvVel2CAN</td></tr><tr><td>トルク</td><td>cnvCur2CAN</td></tr></table> |  | 指令単位 | 変換関数 | 速度        | cnvVel2CAN | トルク  | cnvCur2CAN     |      |   |
| 指令単位      | 変換関数   |  |      |      |           |            |      |                |      |   |
| 速度        | cnvVel2CAN   |  |      |      |           |            |      |                |      |   |
| トルク       | cnvCur2CAN   |  |      |      |           |            |      |                |      |   |



## abh3\_can\_cmd

| 概要        | 指令の送信（同時）  |  |      |      |           |            |       |              |       |         |      |   |
|-----------|--|--|------|------|-----------|------------|-------|--------------|-------|---------|------|---|
| 詳細        | A/Y 指令値と B/X 指令値を同時に送信します。   |  |      |      |           |            |       |              |       |         |      |   |
| 構文        | CANABH3API int32_t abh3_can_cmd(uint8_t nTargetID, int16_t cmdAY, int16_t cmdBX, pCANABH3_RESULT pPtr)   |  |      |      |           |            |       |              |       |         |      |   |
| パラメータ     | <table><tr><th>変数名</th><th>内容</th></tr><tr><td>nTargetID</td><td>送信先 ID</td></tr><tr><td>cmdAY</td><td>A/Y 指令値</td></tr><tr><td>cmdBX</td><td>B/X 指令値</td></tr><tr><td>pPtr</td><td>送信に対する返信の格納領域<br/>NULL を指定すると送信のみで戻ります<br/>受信時は、pPtr-&gt;u. DP0R に格納されます</td></tr></table> |  | 変数名  | 内容   | nTargetID | 送信先 ID     | cmdAY | A/Y 指令値      | cmdBX | B/X 指令値 | pPtr | 送信に対する返信の格納領域<br>NULL を指定すると送信のみで戻ります<br>受信時は、pPtr->u. DP0R に格納されます |
| 変数名       | 内容   |  |      |      |           |            |       |              |       |         |      |   |
| nTargetID | 送信先 ID   |  |      |      |           |            |       |              |       |         |      |   |
| cmdAY     | A/Y 指令値  |  |      |      |           |            |       |              |       |         |      |   |
| cmdBX     | B/X 指令値  |  |      |      |           |            |       |              |       |         |      |   |
| pPtr      | 送信に対する返信の格納領域<br>NULL を指定すると送信のみで戻ります<br>受信時は、pPtr->u. DP0R に格納されます  |  |      |      |           |            |       |              |       |         |      |   |
| 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>0 以外</td><td>異常終了時のエラーコード</td></tr></table>   |  | 戻り値  | 内容   | 0         | 正常終了       | 0 以外  | 異常終了時のエラーコード |       |         |      |   |
| 戻り値       | 内容   |  |      |      |           |            |       |              |       |         |      |   |
| 0         | 正常終了   |  |      |      |           |            |       |              |       |         |      |   |
| 0 以外      | 異常終了時のエラーコード   |  |      |      |           |            |       |              |       |         |      |   |
| 注意点等      | <p>指令として与える値は、適切な変換関数を使用して下さい。<br/>又、制御対象がその指令単位になっている事を先に確認して下さい。</p> <table><tr><th>指令単位</th><th>変換関数</th></tr><tr><td>速度</td><td>cnvVel2CAN</td></tr><tr><td>トルク</td><td>cnvCur2CAN</td></tr></table>   |  | 指令単位 | 変換関数 | 速度        | cnvVel2CAN | トルク   | cnvCur2CAN   |       |         |      |   |
| 指令単位      | 変換関数   |  |      |      |           |            |       |              |       |         |      |   |
| 速度        | cnvVel2CAN   |  |      |      |           |            |       |              |       |         |      |   |
| トルク       | cnvCur2CAN   |  |      |      |           |            |       |              |       |         |      |   |

## abh3\_can\_inSet

|       |   |   |
|-------|---|---|
| 概要    | 入力の送信（一括）   |   |
| 詳細    | 入力(bit 対応)の値をデータ値とマスク値で構築し、送信します。   |   |
| 構文    | CANABH3API int32_t abh3_can_opSet(uint8_t nTargetID, int32_t data, int32_t mask, pCANABH3_RESULT pPtr)  |   |
| パラメータ | 変数名   | 内容  |
|       | nTargetID   | 送信先 ID  |
|       | data  | データ値（各 bit の割り当ては、機器側資料を参照）   |
|       | mask  | マスク値  |
|       | pPtr  | 送信に対する返信の格納領域<br>NULL を指定すると送信のみで戻ります<br>受信時は、pPtr->u. DP0R に格納されます |
| 戻り値   | 戻り値   | 内容  |
|       | 0   | 正常終了  |
|       | 0 以外  | 異常終了時のエラーコード  |
| 注意点等  | <p>実際に送信される値は、以下の様に算出されます。</p> <p>入力(bit 対応) = (入力(bit 対応) &amp; ~mask)   (data &amp; mask)</p> <p>data の値をそのまま送りたい場合は、mask に 0 を指定して下さい。</p> |   |

# ab3\_can\_inBitSet

| 概要        | 入力の送信（ビット）  |     |    |           |        |      |              |      |             |      |   |
|-----------|---|-----|----|-----------|--------|------|--------------|------|-------------|------|---|
| 詳細        | 現在の入力 (bit 対応) の特定ビットを操作し、送信します。  |     |    |           |        |      |              |      |             |      |   |
| 構文        | CANABH3API int32_t ab3_can_inBitSet(uint8_t nTargetID, int8_t num, int8_t data, pCANABH3_RESULT pPtr)   |     |    |           |        |      |              |      |             |      |   |
| パラメータ     | <table> <tr> <th>変数名</th><th>内容</th></tr> <tr> <td>nTargetID</td><td>送信先 ID</td></tr> <tr> <td>num</td><td>ビット番号 (0～31)</td></tr> <tr> <td>data</td><td>設定データ (0～1)</td></tr> <tr> <td>pPtr</td><td>送信に対する返信の格納領域<br/>NULL を指定すると送信のみで戻ります<br/>受信時は、pPtr-&gt;u. DP0R に格納されます</td></tr> </table> | 変数名 | 内容 | nTargetID | 送信先 ID | num  | ビット番号 (0～31) | data | 設定データ (0～1) | pPtr | 送信に対する返信の格納領域<br>NULL を指定すると送信のみで戻ります<br>受信時は、pPtr->u. DP0R に格納されます |
| 変数名       | 内容  |     |    |           |        |      |              |      |             |      |   |
| nTargetID | 送信先 ID  |     |    |           |        |      |              |      |             |      |   |
| num       | ビット番号 (0～31)  |     |    |           |        |      |              |      |             |      |   |
| data      | 設定データ (0～1)   |     |    |           |        |      |              |      |             |      |   |
| pPtr      | 送信に対する返信の格納領域<br>NULL を指定すると送信のみで戻ります<br>受信時は、pPtr->u. DP0R に格納されます   |     |    |           |        |      |              |      |             |      |   |
| 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>0 以外</td><td>異常終了時のエラーコード</td></tr> </table>   | 戻り値 | 内容 | 0         | 正常終了   | 0 以外 | 異常終了時のエラーコード |      |             |      |   |
| 戻り値       | 内容  |     |    |           |        |      |              |      |             |      |   |
| 0         | 正常終了  |     |    |           |        |      |              |      |             |      |   |
| 0 以外      | 異常終了時のエラーコード  |     |    |           |        |      |              |      |             |      |   |
| 注意点等      | <p>実際に送信される値は、以下の様に算出されます。</p> <p>入力 (bit 対応) = 入力 (bit 対応) &amp; ~(1 &lt;&lt; num)   (data &lt;&lt; num)</p>   |     |    |           |        |      |              |      |             |      |   |

# ab33\_can\_cmdAndopSet

| 概要        | 指令と入力の送信（一括）  |     |    |           |        |       |              |       |         |      |      |      |      |      |   |
|-----------|---|-----|----|-----------|--------|-------|--------------|-------|---------|------|------|------|------|------|---|
| 詳細        | 指令 (A/Y, B/X) と、入力 (bit 対応) の値をデータ値とマスク値で構築し、送信します。   |     |    |           |        |       |              |       |         |      |      |      |      |      |   |
| 構文        | CANABH3API int32_t ab33_can_cmdAndopSet(uint8_t nTargetID,<br>int16_t cmdAY, int16_t cmdBX,<br>int32_t data, int32_t mask,<br>pCANABH3_RESULT pPtr)   |     |    |           |        |       |              |       |         |      |      |      |      |      |   |
| パラメータ     | <table> <tr> <th>変数名</th><th>内容</th></tr> <tr> <td>nTargetID</td><td>送信先 ID</td></tr> <tr> <td>cmdAY</td><td>A/Y 指令値</td></tr> <tr> <td>cmdBX</td><td>B/X 指令値</td></tr> <tr> <td>data</td><td>データ値</td></tr> <tr> <td>mask</td><td>マスク値</td></tr> <tr> <td>pPtr</td><td>送信に対する返信の格納領域<br/>NULL を指定すると送信のみで戻ります<br/>受信時は、pPtr-&gt;u. DPOR に格納されます</td></tr> </table> | 変数名 | 内容 | nTargetID | 送信先 ID | cmdAY | A/Y 指令値      | cmdBX | B/X 指令値 | data | データ値 | mask | マスク値 | pPtr | 送信に対する返信の格納領域<br>NULL を指定すると送信のみで戻ります<br>受信時は、pPtr->u. DPOR に格納されます |
| 変数名       | 内容  |     |    |           |        |       |              |       |         |      |      |      |      |      |   |
| nTargetID | 送信先 ID  |     |    |           |        |       |              |       |         |      |      |      |      |      |   |
| cmdAY     | A/Y 指令値   |     |    |           |        |       |              |       |         |      |      |      |      |      |   |
| cmdBX     | B/X 指令値   |     |    |           |        |       |              |       |         |      |      |      |      |      |   |
| data      | データ値  |     |    |           |        |       |              |       |         |      |      |      |      |      |   |
| mask      | マスク値  |     |    |           |        |       |              |       |         |      |      |      |      |      |   |
| pPtr      | 送信に対する返信の格納領域<br>NULL を指定すると送信のみで戻ります<br>受信時は、pPtr->u. DPOR に格納されます   |     |    |           |        |       |              |       |         |      |      |      |      |      |   |
| 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>0 以外</td><td>異常終了時のエラーコード</td></tr> </table>   | 戻り値 | 内容 | 0         | 正常終了   | 0 以外  | 異常終了時のエラーコード |       |         |      |      |      |      |      |   |
| 戻り値       | 内容  |     |    |           |        |       |              |       |         |      |      |      |      |      |   |
| 0         | 正常終了  |     |    |           |        |       |              |       |         |      |      |      |      |      |   |
| 0 以外      | 異常終了時のエラーコード  |     |    |           |        |       |              |       |         |      |      |      |      |      |   |
| 注意点等      | <p>本関数は ab33_can_cmd 関数と ab33_can_opSet 関数の処理を一度に行う為の関数です。</p> <p>指令は ab33_can_cmd 関数、入力は ab33_can_opSet 関数に準拠する為、説明は各関数の項目を参照願います。</p>   |     |    |           |        |       |              |       |         |      |      |      |      |      |   |

## ab3h3\_can\_reqBRD

| 概要                       | ブロードキャストパケットのリクエスト  |  |                          |     |           |              |      |   |      |  |   |              |   |              |   |              |   |              |
|--------------------------|---|--|--------------------------|-----|-----------|--------------|------|---|------|--|---|--------------|---|--------------|---|--------------|---|--------------|
| 詳細                       | 指定番号のブロードキャストパケットを送信し、指定番号に対する要素を取得します。   |  |                          |     |           |              |      |   |      |  |   |              |   |              |   |              |   |              |
| 構文                       | CANABH3API int32_t abh3_can_reqBRD(uint8_t nTargetID, uint8_t num, pCANABH3_RESULT pPtr)  |  |                          |     |           |              |      |   |      |  |   |              |   |              |   |              |   |              |
| パラメータ                    | <table><tr><th>変数名</th><th>内容</th></tr><tr><td>nTargetID</td><td>送信先 ID</td></tr><tr><td>num</td><td>番号 (0x00~0xff)<br/>下位 3bit には機器アドレス<br/>上位 5bit にはグループ番号が入ります</td></tr><tr><td>pPtr</td><td>送信に対する返信の格納領域<br/>NULL を指定すると送信のみで戻ります<br/>受信時の格納先は、本関数の注意点等を参照。</td></tr></table>   |  | 変数名                      | 内容  | nTargetID | 送信先 ID       | num  | 番号 (0x00~0xff)<br>下位 3bit には機器アドレス<br>上位 5bit にはグループ番号が入ります | pPtr | 送信に対する返信の格納領域<br>NULL を指定すると送信のみで戻ります<br>受信時の格納先は、本関数の注意点等を参照。 |   |              |   |              |   |              |   |              |
| 変数名                      | 内容  |  |                          |     |           |              |      |   |      |  |   |              |   |              |   |              |   |              |
| nTargetID                | 送信先 ID  |  |                          |     |           |              |      |   |      |  |   |              |   |              |   |              |   |              |
| num                      | 番号 (0x00~0xff)<br>下位 3bit には機器アドレス<br>上位 5bit にはグループ番号が入ります   |  |                          |     |           |              |      |   |      |  |   |              |   |              |   |              |   |              |
| pPtr                     | 送信に対する返信の格納領域<br>NULL を指定すると送信のみで戻ります<br>受信時の格納先は、本関数の注意点等を参照。  |  |                          |     |           |              |      |   |      |  |   |              |   |              |   |              |   |              |
| 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>0 以外</td><td>異常終了時のエラーコード</td></tr></table>  |  | 戻り値                      | 内容  | 0         | 正常終了         | 0 以外 | 異常終了時のエラーコード  |      |  |   |              |   |              |   |              |   |              |
| 戻り値                      | 内容  |  |                          |     |           |              |      |   |      |  |   |              |   |              |   |              |   |              |
| 0                        | 正常終了  |  |                          |     |           |              |      |   |      |  |   |              |   |              |   |              |   |              |
| 0 以外                     | 異常終了時のエラーコード  |  |                          |     |           |              |      |   |      |  |   |              |   |              |   |              |   |              |
| 注意点等                     | <p>パラメータの num に対する通信結果を受け取る領域は以下の通り。<br/>内容に関しては、構造体の説明を参照の事。</p> <table><tr><th>機器アドレス<br/>(num の下位 3bit)</th><th>格納先</th></tr><tr><td>0</td><td>pPtr-&gt;u. BR0</td></tr><tr><td>1</td><td>pPtr-&gt;u. BR1</td></tr><tr><td>2</td><td>pPtr-&gt;u. BR2</td></tr><tr><td>3</td><td>pPtr-&gt;u. BR3</td></tr><tr><td>4</td><td>pPtr-&gt;u. BR4</td></tr><tr><td>5</td><td>pPtr-&gt;u. BR5</td></tr><tr><td>6</td><td>pPtr-&gt;u. BR6</td></tr></table> |  | 機器アドレス<br>(num の下位 3bit) | 格納先 | 0         | pPtr->u. BR0 | 1    | pPtr->u. BR1  | 2    | pPtr->u. BR2   | 3 | pPtr->u. BR3 | 4 | pPtr->u. BR4 | 5 | pPtr->u. BR5 | 6 | pPtr->u. BR6 |
| 機器アドレス<br>(num の下位 3bit) | 格納先   |  |                          |     |           |              |      |   |      |  |   |              |   |              |   |              |   |              |
| 0                        | pPtr->u. BR0  |  |                          |     |           |              |      |   |      |  |   |              |   |              |   |              |   |              |
| 1                        | pPtr->u. BR1  |  |                          |     |           |              |      |   |      |  |   |              |   |              |   |              |   |              |
| 2                        | pPtr->u. BR2  |  |                          |     |           |              |      |   |      |  |   |              |   |              |   |              |   |              |
| 3                        | pPtr->u. BR3  |  |                          |     |           |              |      |   |      |  |   |              |   |              |   |              |   |              |
| 4                        | pPtr->u. BR4  |  |                          |     |           |              |      |   |      |  |   |              |   |              |   |              |   |              |
| 5                        | pPtr->u. BR5  |  |                          |     |           |              |      |   |      |  |   |              |   |              |   |              |   |              |
| 6                        | pPtr->u. BR6  |  |                          |     |           |              |      |   |      |  |   |              |   |              |   |              |   |              |

## abh3\_can\_trans

| 概要        | 非公開  |  |     |    |           |        |      |              |      |           |         |                     |
|-----------|--|--|-----|----|-----------|--------|------|--------------|------|-----------|---------|---------------------|
| 詳細        | 非公開  |  |     |    |           |        |      |              |      |           |         |                     |
| 構文        | CANABH3API int32_t abh3_can_trans(uint8_t nTargetID, char* sbuf, char* rbuf, uint32_t* rbuflen)  |  |     |    |           |        |      |              |      |           |         |                     |
| パラメータ     | <table><tr><th>変数名</th><th>内容</th></tr><tr><td>nTargetID</td><td>送信先 ID</td></tr><tr><td>sbuf</td><td>送信データバッファ</td></tr><tr><td>rbuf</td><td>受信データバッファ</td></tr><tr><td>rbuflen</td><td>受信データバッファサイズ[bytes]</td></tr></table> |  | 変数名 | 内容 | nTargetID | 送信先 ID | sbuf | 送信データバッファ    | rbuf | 受信データバッファ | rbuflen | 受信データバッファサイズ[bytes] |
| 変数名       | 内容   |  |     |    |           |        |      |              |      |           |         |                     |
| nTargetID | 送信先 ID   |  |     |    |           |        |      |              |      |           |         |                     |
| sbuf      | 送信データバッファ  |  |     |    |           |        |      |              |      |           |         |                     |
| rbuf      | 受信データバッファ  |  |     |    |           |        |      |              |      |           |         |                     |
| rbuflen   | 受信データバッファサイズ[bytes]  |  |     |    |           |        |      |              |      |           |         |                     |
| 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>0 以外</td><td>異常終了時のエラーコード</td></tr></table>   |  | 戻り値 | 内容 | 0         | 正常終了   | 0 以外 | 異常終了時のエラーコード |      |           |         |                     |
| 戻り値       | 内容   |  |     |    |           |        |      |              |      |           |         |                     |
| 0         | 正常終了   |  |     |    |           |        |      |              |      |           |         |                     |
| 0 以外      | 異常終了時のエラーコード   |  |     |    |           |        |      |              |      |           |         |                     |
| 注意点等      | 本関数は特殊なコマンドをマルチパケットで送る為の関数です。<br>パラメータで指定される受信データバッファは、十分な容量が必要です。   |  |     |    |           |        |      |              |      |           |         |                     |

## abh3\_can\_copylastdata

| 概要        | 最終受信データをコピー   |  |     |    |           |                 |      |                |
|-----------|---|--|-----|----|-----------|-----------------|------|----------------|
| 詳細        | CAN 回線からの受信データを ID 毎に格納した領域をコピーします。   |  |     |    |           |                 |      |                |
| 構文        | CANABH3API int32_t abh3_can_copylastdata(uint8_t nTargetID, pCANABH3_LASTRECV pPtr)   |  |     |    |           |                 |      |                |
| パラメータ     | <table><tr><th>変数名</th><th>内容</th></tr><tr><td>nTargetID</td><td>コピー対象のデータ発信元 ID</td></tr><tr><td>pPtr</td><td>最終受信データを格納する領域</td></tr></table> |  | 変数名 | 内容 | nTargetID | コピー対象のデータ発信元 ID | pPtr | 最終受信データを格納する領域 |
| 変数名       | 内容  |  |     |    |           |                 |      |                |
| nTargetID | コピー対象のデータ発信元 ID   |  |     |    |           |                 |      |                |
| pPtr      | 最終受信データを格納する領域  |  |     |    |           |                 |      |                |
| 戻り値       | 無し  |  |     |    |           |                 |      |                |
| 注意点等      | abh3_can_flush 関数を呼び出した時点の格納済みデータがコピーされます。<br>使用する構造体は、シングルパケット・ブロードキャストパケットが別々の領域として用意されています。<br>詳細は、「非対称通信について」の項を参照願います。                     |  |     |    |           |                 |      |                |

## abh3\_can\_resetlastdata

|       |   |                        |      |
|-------|---|------------------------|------|
| 概要    | 最終受信データの指定更新フラグを解除  |                        |      |
| 詳細    | <p>最終受信データ指定箇所の更新フラグを解除します。</p> <p>最終受信データは、「受信データの格納場所に対する更新フラグ」が成立し、データの受信場所が判明する仕様になっていますが、この更新フラグを解除します。更新フラグはこの関数で解除しない限り、解除されません。</p> |                        |      |
| 構文    | CANABH3API int32_t abh3_can_resetlastdata(uint8_t nTargetID, int32_t nAdrs)   |                        |      |
| パラメータ | 変数名   | 内容                     |      |
|       | nTargetID   | 対象の発信元 ID              |      |
|       | nAdrs   | 最終受信データの更新フラグを解除するアドレス |      |
|       |   | 値                      | 解除場所 |
|       |   | 0                      | DPOR |
|       |   | 1                      | BR0  |
|       |   | 2                      | BR1  |
|       |   | 3                      | BR2  |
|       |   | 4                      | BR3  |
|       |   | 5                      | BR4  |
|       |   | 6                      | BR5  |
|       |   | 7                      | BR6  |
| 戻り値   | 無し  |                        |      |
| 注意点等  | 利用法や詳細は、「非対称通信について」の項を参照願います。   |                        |      |



## abh3\_can\_read

| 概要               | 指定種類のパケットを受信する   |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
|------------------|--|-----|---------|---------------|----------|------------------|----------------------------|------------|--|---|----------|---------------|----------|------------------|------------------|------------|------------|-------|---|---|--------|---|----------|---|----------|---|----------|---|----------|---|----------|---|----------|---|----------|
| 詳細               | 指定相手先が送信した指定種類のパケットを受信する。<br>指定種類以外のパケットを受信した場合は、無視されます。<br>(最終更新データには格納されますが、処理は戻りません)  |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| 構文               | CANABH3API int32_t abh3_can_read(uint8_t nTargetID, pCANABH3_RESULT pPtr,<br>PACKETTYPE nType, uint8_t nAdrs)  |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| パラメータ            | <table border="1"> <thead> <tr> <th>変数名</th><th>内容</th></tr> </thead> <tbody> <tr> <td>nTargetID</td><td>発信元 ID</td></tr> <tr> <td>pPtr</td><td>送信に対する返信の格納領域<br/>NULL 指定不可</td></tr> <tr> <td>nType</td><td>受信するパケットタイプを指定する <table border="1"> <thead> <tr> <th>値</th><th>パケットタイプ</th></tr> </thead> <tbody> <tr> <td>SINGLE_PACKET</td><td>シングルパケット</td></tr> <tr> <td>BROADCAST_PACKET</td><td>ブロードキャスト<br/>パケット</td></tr> <tr> <td>ANY_PACKET</td><td>パケット種類問わない</td></tr> </tbody> </table> </td></tr> <tr> <td>nAdrs</td><td>nType がブロードキャストパケット指定時に受信対象の<br/>機器アドレスを指定する。<br/>他のパケット種類では本パラメータは無視される。 <table border="1"> <thead> <tr> <th>値</th><th>機器アドレス</th></tr> </thead> <tbody> <tr> <td>0</td><td>機器アドレス 0</td></tr> <tr> <td>1</td><td>機器アドレス 1</td></tr> <tr> <td>2</td><td>機器アドレス 2</td></tr> <tr> <td>3</td><td>機器アドレス 3</td></tr> <tr> <td>4</td><td>機器アドレス 4</td></tr> <tr> <td>5</td><td>機器アドレス 5</td></tr> <tr> <td>6</td><td>機器アドレス 6</td></tr> </tbody> </table> </td></tr> </tbody> </table> | 変数名 | 内容      | nTargetID     | 発信元 ID   | pPtr             | 送信に対する返信の格納領域<br>NULL 指定不可 | nType      | 受信するパケットタイプを指定する <table border="1"> <thead> <tr> <th>値</th><th>パケットタイプ</th></tr> </thead> <tbody> <tr> <td>SINGLE_PACKET</td><td>シングルパケット</td></tr> <tr> <td>BROADCAST_PACKET</td><td>ブロードキャスト<br/>パケット</td></tr> <tr> <td>ANY_PACKET</td><td>パケット種類問わない</td></tr> </tbody> </table> | 値 | パケットタイプ  | SINGLE_PACKET | シングルパケット | BROADCAST_PACKET | ブロードキャスト<br>パケット | ANY_PACKET | パケット種類問わない | nAdrs | nType がブロードキャストパケット指定時に受信対象の<br>機器アドレスを指定する。<br>他のパケット種類では本パラメータは無視される。 <table border="1"> <thead> <tr> <th>値</th><th>機器アドレス</th></tr> </thead> <tbody> <tr> <td>0</td><td>機器アドレス 0</td></tr> <tr> <td>1</td><td>機器アドレス 1</td></tr> <tr> <td>2</td><td>機器アドレス 2</td></tr> <tr> <td>3</td><td>機器アドレス 3</td></tr> <tr> <td>4</td><td>機器アドレス 4</td></tr> <tr> <td>5</td><td>機器アドレス 5</td></tr> <tr> <td>6</td><td>機器アドレス 6</td></tr> </tbody> </table> | 値 | 機器アドレス | 0 | 機器アドレス 0 | 1 | 機器アドレス 1 | 2 | 機器アドレス 2 | 3 | 機器アドレス 3 | 4 | 機器アドレス 4 | 5 | 機器アドレス 5 | 6 | 機器アドレス 6 |
| 変数名              | 内容   |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| nTargetID        | 発信元 ID   |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| pPtr             | 送信に対する返信の格納領域<br>NULL 指定不可   |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| nType            | 受信するパケットタイプを指定する <table border="1"> <thead> <tr> <th>値</th><th>パケットタイプ</th></tr> </thead> <tbody> <tr> <td>SINGLE_PACKET</td><td>シングルパケット</td></tr> <tr> <td>BROADCAST_PACKET</td><td>ブロードキャスト<br/>パケット</td></tr> <tr> <td>ANY_PACKET</td><td>パケット種類問わない</td></tr> </tbody> </table>   | 値   | パケットタイプ | SINGLE_PACKET | シングルパケット | BROADCAST_PACKET | ブロードキャスト<br>パケット           | ANY_PACKET | パケット種類問わない   |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| 値                | パケットタイプ  |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| SINGLE_PACKET    | シングルパケット   |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| BROADCAST_PACKET | ブロードキャスト<br>パケット   |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| ANY_PACKET       | パケット種類問わない   |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| nAdrs            | nType がブロードキャストパケット指定時に受信対象の<br>機器アドレスを指定する。<br>他のパケット種類では本パラメータは無視される。 <table border="1"> <thead> <tr> <th>値</th><th>機器アドレス</th></tr> </thead> <tbody> <tr> <td>0</td><td>機器アドレス 0</td></tr> <tr> <td>1</td><td>機器アドレス 1</td></tr> <tr> <td>2</td><td>機器アドレス 2</td></tr> <tr> <td>3</td><td>機器アドレス 3</td></tr> <tr> <td>4</td><td>機器アドレス 4</td></tr> <tr> <td>5</td><td>機器アドレス 5</td></tr> <tr> <td>6</td><td>機器アドレス 6</td></tr> </tbody> </table>  | 値   | 機器アドレス  | 0             | 機器アドレス 0 | 1                | 機器アドレス 1                   | 2          | 機器アドレス 2   | 3 | 機器アドレス 3 | 4             | 機器アドレス 4 | 5                | 機器アドレス 5         | 6          | 機器アドレス 6   |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| 値                | 機器アドレス   |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| 0                | 機器アドレス 0   |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| 1                | 機器アドレス 1   |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| 2                | 機器アドレス 2   |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| 3                | 機器アドレス 3   |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| 4                | 機器アドレス 4   |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| 5                | 機器アドレス 5   |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| 6                | 機器アドレス 6   |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| 戻り値              | <table border="1"> <thead> <tr> <th>戻り値</th><th>内容</th></tr> </thead> <tbody> <tr> <td>0</td><td>正常終了</td></tr> <tr> <td>0 以外</td><td>異常終了時のエラーコード</td></tr> </tbody> </table>   | 戻り値 | 内容      | 0             | 正常終了     | 0 以外             | 異常終了時のエラーコード               |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| 戻り値              | 内容   |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| 0                | 正常終了   |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| 0 以外             | 異常終了時のエラーコード   |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |
| 注意点等             | <p>本関数を呼び出すと、</p> <ul style="list-style-type: none"> <li>・ 指定発信元から指定パケットに合致する物を受信</li> <li>・ 受信タイムアウト</li> </ul> <p>どちらかの状態になるまで処理が呼び出し元へ戻らないので注意。<br/>条件に合う/合わない関係なく、パケットを受信した場合は、最終受信データに反映される。</p>  |     |         |               |          |                  |                            |            |  |   |          |               |          |                  |                  |            |            |       |   |   |        |   |          |   |          |   |          |   |          |   |          |   |          |   |          |

## abh3\_can\_flush

| 概要    | 受信バッファを空にします   |  |     |    |   |      |      |              |
|-------|--|--|-----|----|---|------|------|--------------|
| 詳細    | 受信バッファにある内容を全て受信し、最終受信データに反映させます。<br>受信により上書きされた最終受信データ領域は、更新フラグが成立します。  |  |     |    |   |      |      |              |
| 構文    | CANABH3API int32_t abh3_can_flush()  |  |     |    |   |      |      |              |
| パラメータ | 無し   |  |     |    |   |      |      |              |
| 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>0 以外</td><td>異常終了時のエラーコード</td></tr></table> |  | 戻り値 | 内容 | 0 | 正常終了 | 0 以外 | 異常終了時のエラーコード |
| 戻り値   | 内容   |  |     |    |   |      |      |              |
| 0     | 正常終了   |  |     |    |   |      |      |              |
| 0 以外  | 異常終了時のエラーコード   |  |     |    |   |      |      |              |
| 注意点等  | 利用法や詳細は、「非対称通信について」の項を参照願います。  |  |     |    |   |      |      |              |

## abh3\_can\_finish

| 概要    | CAN 回線の切断  |  |     |    |   |      |      |              |
|-------|--|--|-----|----|---|------|------|--------------|
| 詳細    | CAN 回線から切断します。   |  |     |    |   |      |      |              |
| 構文    | CANABH3API int32_t abh3_can_finish()   |  |     |    |   |      |      |              |
| パラメータ | 無し   |  |     |    |   |      |      |              |
| 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>0 以外</td><td>異常終了時のエラーコード</td></tr></table> |  | 戻り値 | 内容 | 0 | 正常終了 | 0 以外 | 異常終了時のエラーコード |
| 戻り値   | 内容   |  |     |    |   |      |      |              |
| 0     | 正常終了   |  |     |    |   |      |      |              |
| 0 以外  | 異常終了時のエラーコード   |  |     |    |   |      |      |              |
| 注意点等  |  |  |     |    |   |      |      |              |

# cnvVel2CAN

|       |   |                             |
|-------|---|-----------------------------|
| 概要    | 速度を「ABH3 の速度」に変換                                    |                             |
| 詳細    | ユーザーが扱う [min <sup>-1</sup> ] の速度値を、「ABH3 で扱う速度」に変換。 |                             |
| 構文    | CANABH3API int16_t cnvVel2CAN(float vel)            |                             |
| パラメータ |   |                             |
|       | 変数名   | 内容                          |
|       | vel   | 変換元の速度 [min <sup>-1</sup> ] |
| 戻り値   | 変換された速度が戻ります。                                       |                             |
| 注意点等  | ユーザー側で扱う値と ABH3 で扱う値の関係は、「値の単位」の項を参照の事。             |                             |

# cnvCAN2Vel

|       |  |       |
|-------|--|-------|
| 概要    | 「ABH3 の速度」を変換                                      |       |
| 詳細    | 「ABH3 で扱う速度」の値を、ユーザーが扱う速度[ $\text{min}^{-1}$ ]に変換。 |       |
| 構文    | CANABH3API float cnvCAN2Vel(int16_t vel)           |       |
| パラメータ |  |       |
|       | 変数名  | 内容    |
|       | vel  | 変換元の値 |
| 戻り値   | 変換された速度[ $\text{min}^{-1}$ ]が戻ります。                 |       |
| 注意点等  | ユーザー側で扱う値と ABH3 で扱う値の関係は、「値の単位」の項を参照の事。            |       |

# cnvCur2CAN

|       |   |       |
|-------|---|-------|
| 概要    | 電流値を「ABH3 の電流値」に変換                      |       |
| 詳細    | ユーザーが扱う電流値[%]を、「ABH3 で扱う電流値」に変換。        |       |
| 構文    | CANABH3API float cnvCur2CAN(float cur)  |       |
| パラメータ |   |       |
|       | 変数名                                     | 内容    |
|       | cur                                     | 変換元の値 |
| 戻り値   | 変換された電流値[%]が戻ります。                       |       |
| 注意点等  | ユーザー側で扱う値と ABH3 で扱う値の関係は、「値の単位」の項を参照の事。 |       |

# cnvCAN2Cur

|       |   |       |
|-------|---|-------|
| 概要    | 「ABH3 の電流値」を変換                            |       |
| 詳細    | 「ABH3 で扱う電流値」の値を、ユーザーが扱う電流値[%]に変換。        |       |
| 構文    | CANABH3API float cnvCAN2Cur (int16_t cur) |       |
| パラメータ | 変数名                                       | 内容    |
|       | cur                                       | 変換元の値 |
| 戻り値   | 変換された電流値が戻ります。                            |       |
| 注意点等  | ユーザー側で扱う値と ABH3 で扱う値の関係は、「値の単位」の項を参照の事。   |       |

# cnvCAN2Load

|       |  |       |
|-------|--|-------|
| 概要    | 「ABH3 の負荷率」を変換                             |       |
| 詳細    | 「ABH3 で扱う負荷率」の値を、ユーザーが扱う負荷率[%]に変換。         |       |
| 構文    | CANABH3API float cnvCAN2Load(int16_t load) |       |
| パラメータ |  |       |
|       | 変数名  | 内容    |
|       | load                                       | 変換元の値 |
| 戻り値   | 変換された負荷率[%]が戻ります。                          |       |
| 注意点等  | ユーザー側で扱う値と ABH3 で扱う値の関係は、「値の単位」の項を参照の事。    |       |



## cnvCAN2Analog

|       |  |       |
|-------|--|-------|
| 概要    | 「ABH3 のアナログ入力」を変換                              |       |
| 詳細    | 「ABH3 で扱うアナログ入力」の値を、ユーザーが扱うアナログ入力値[V]に変換。      |       |
| 構文    | CANABH3API float cnvCAN2Analog(int16_t analog) |       |
| パラメータ |  |       |
|       | 変数名  | 内容    |
|       | analog   | 変換元の値 |
| 戻り値   | 変換されたアナログ入力値[V]が戻ります。                          |       |
| 注意点等  | ユーザー側で扱う値と ABH3 で扱う値の関係は、「値の単位」の項を参照の事。        |       |

# cnvCAN2Volt

|       |  |       |
|-------|--|-------|
| 概要    | 「ABH3 の電源電圧値」を変換                           |       |
| 詳細    | 「ABH3 で扱う電源電圧値」の値を、ユーザーが扱う電源電圧値[V]に変換。     |       |
| 構文    | CANABH3API float cnvCAN2Volt(int16_t volt) |       |
| パラメータ |  |       |
|       | 変数名  | 内容    |
|       | volt                                       | 変換元の値 |
| 戻り値   | 変換された電源電圧値[V]が戻ります。                        |       |
| 注意点等  | ユーザー側で扱う値と ABH3 で扱う値の関係は、「値の単位」の項を参照の事。    |       |

## 値の単位

ユーザー側で使用する単位と ABH3 側に指定する値の関係は以下の通り

| 要素         | ユーザー側単位 | ABH3 分解能  | 変換関数        |               |
|------------|---------|-----------|-------------|---------------|
|            |         |           | ユーザー → ABH3 | ABH3 → ユーザー   |
| 速度（指令・帰還）  | [min]   | 0.2 [min] | cnvVel2CAN  | cnvCAN2Vel    |
| 電流（指令・帰還）  | [%]     | 0.01 [%]  | cnvCur2CAN  | cnvCAN2Cur    |
| パルス積算値     | [Pulse] | 1 [Pulse] | —           | —             |
| 負荷率        | [%]     | 1 [%]     | —           | cnvCAN2Load   |
| 主電源／制御電源電圧 | [V]     | 0.1 [V]   | —           | cnvCAN2VOLT   |
| アナログ入力     | [V]     | 0.01 [V]  | —           | cnvCAN2Analog |
| モニタデータ     | 単位無し    | 単位無し      | —           | —             |

## 非対称通信について

本 DLL が想定している接続先との通信は、以下の様な方法があり、条件により非対称(\*1)の場合があります。

| No. | 挙動  | 通信方式  |
|-----|---|-------|
| 1   | PC 側から送信すると、必ず送信先から返信が届く。   | 通常    |
| 2   | PC 側から送信しても、送信先から返信は来ない。  | 非対称 1 |
| 3   | PC 側から送信しなくても、返信に相当する物が届く。<br>PC 側から送信した場合の返答は、有り無しどちらにも対応が必要な場合。 | 非対称 2 |

通信方式により本 DLL を以下の様に使用する事が推奨されます。

| 通信方式  | 通信方法と利用条件等  |
|-------|---|
| 通常    | 通信用関数 (abh3_can_*) では、受信領域を指定して利用する。<br>送信と受信をセットで扱う事になる。   |
| 非対称 1 | 通信用関数 (abh3_can_*) では、送信処理のみとする為、受信領域に NULL 指定。<br>送信だけ行う事になる。  |
| 非対称 2 | 通信用関数 (abh3_can_*) では、送信処理のみとする為、受信領域に NULL 指定。<br>受信は勝手に送られてくる分も含めて処理する為、一定間隔で abh3_can_flush 関数を呼び出して最終受信データとして内部に取り込む必要が有る。<br><br>それと同じタイミング、又は別のタイミングで、受信した内容の処理を行う為に<br>(1) abh3_can_copylastdata 関数で最終受信データを取得<br>(2) 取得した構造体の update メンバが成立 (0 以外の値) している所を確認<br>(3) 該当箇所を指定して、abh3_can_resetlastdata 関数を呼び出す<br>(4) 該当箇所のデータを元に、必要な処理が有れば処理する。<br>(5) これを全ての update メンバに対して行う ((2)に戻る)<br>この様な処理となる。 |

(\*1) PC(ホスト)が送信しても、機器側が無返信、又は機器側が無条件で PC(ホスト)に何か送ってくる場合。  
主に送信と受信が 1:1 になっていない関係を指す。

## 実装例

### 以下前提条件に対する実装の例

#### 前提条件

| No. | 条件   |
|-----|--|
| (1) | PC 側は一定周期毎に指令を送信する必要がある                      |
| (2) | PC 側はブロードキャストで得られる値（各種ステータス類）を一定周期で表示する必要がある |
| (3) | ABH3 側がブロードキャスト(0～6)で全ての項目を自動配信する設定にしてある     |
| (4) | ABH3 側はシングルパケットに対して必ず返答する                    |

#### 上記条件に合う実装の例

| No. | 元条件と実装された処理   |
|-----|---|
| 1   | <p>元条件</p> <p>(1) PC 側は一定周期毎に指令を送信する必要がある</p> <p>(4) ABH3 側はシングルパケットに対して必ず返答する</p> <p>解説</p> <p>指令を送るシングルパケットで必ず返答がある条件の為、送信→受信の処理をセットで行う。</p> <p>実装例</p> <p>一定周期毎に指令と入力の一括送信 (abh3_can_cmdAndopSet 関数) を受信領域指定有りで使用</p>  |
| 2   | <p>元条件</p> <p>(2) PC 側はブロードキャストで得られる値（各種ステータス類）を一定周期で表示する必要がある</p> <p>(3) ABH3 側がブロードキャスト(0～6)で全ての項目を自動配信する設定にしてある</p> <p>解説</p> <p>PC 側から要求しなくても勝手に送ってくるので、PC 側は受信したデータを格納する処理を表示周期で行い、格納結果を表示に利用するだけで良い処理となる。</p> <p>実装例</p> <p>表示周期毎に以下処理を実行する</p> <p>(1) abh3_can_flush 関数を呼び出して受信バッファ内の物を処理させる</p> <p>(2) abh3_can_copylastdata 関数で最終受信データを取得する</p> <p>(3) 取得した構造体に含まれる値を表示する（要変換）</p> |