

## ABH3 CAN 通信 ホスト用サンプルソフト仕様

1	パラメータ .....	2
2	CAN プロトコル概要 .....	3
2.1	シングルパケット (DP=0) .....	3
2.2	シングルパケット (DP=1) .....	3
2.3	ブロードキャストパケット (0x28) .....	3
2.4	ブロードキャストパケット (0x29) .....	3
2.5	ブロードキャストパケット (0x2a) .....	4
2.6	ブロードキャストパケット (0x2b) .....	4
2.7	ブロードキャストパケット (0x2c) .....	4
2.8	ブロードキャストパケット (0x2d) .....	4
2.9	ブロードキャストパケット (0x2e) .....	4
3	データ型 .....	5
4	関数定義 .....	6
4.1	指令の初期化 .....	6
4.2	指令の送信 (軸別) .....	6
4.3	指令の送信 (同時) .....	7
4.4	入力の送信 (一括) .....	7
4.5	入力の送信 (ビット) .....	8
4.6	積算値のリクエスト .....	8
4.7	ブロードキャストパケットのリクエスト .....	9
4.8	各種変換用関数 .....	10
4.9	戻り値の構造体 (共用体) .....	10

## 1 パラメータ

サンプルソフトの動作条件を固定するため、ABH3 の CAN 用パラメータは以下の設定とする。

名称	設定値	備考
CAN 機能	有効	
ボーレート	250/500/1000[kbps]	任意であり、ホスト機器の上限値とする。
ABH3 アドレス	1	
ホスト機器アドレス	2	
シングルパケット 0 送信方式	-1：受信時応答	ABH3 は受信時に応答する
シングルパケット 1 送信方式	-1：受信時応答	ABH3 は受信時に応答する
ブロードキャスト送信グループ番号	5	
ブロードキャスト 0 送信方式	0：リクエストのみ	ABH3 はリクエストに応答する
ブロードキャスト 1 送信方式	0：リクエストのみ	ABH3 はリクエストに応答する
ブロードキャスト 2 送信方式	0：リクエストのみ	ABH3 はリクエストに応答する
ブロードキャスト 3 送信方式	0：リクエストのみ	ABH3 はリクエストに応答する
ブロードキャスト 4 送信方式	0：リクエストのみ	ABH3 はリクエストに応答する
ブロードキャスト 5 送信方式	0：リクエストのみ	ABH3 はリクエストに応答する
ブロードキャスト 6 送信方式	0：リクエストのみ	ABH3 はリクエストに応答する
ブロードキャスト 7 送信方式	0：リクエストのみ	ABH3 はリクエストに応答する
ブロードキャスト受信グループ番号 1	なし	
ブロードキャスト受信グループ番号 2	なし	
ブロードキャスト受信グループ番号 3	なし	
ブロードキャスト受信グループ番号 4	なし	
シングルパケット優先順位	0	
ブロードキャストパケット優先順位	0	
指令ゼロ タイムアウト判定値	500	
異常判定 タイムアウト判定値	1000	

## 2 CAN プロトコル概要

### 2.1 シングルパケット(DP=0)

内容	CAN-ID	DATA							
		1	2	3	4	5	6	7	8
ホスト機器から ABH3	0x00ef0102	A/Y 指令		B/X 指令		入力(bit 対応)			
ABH3 からホスト機器	0x00ef0201	A 速度帰還		B 速度帰還		Y 速度帰還		X 速度帰還	

### 2.2 シングルパケット(DP=1)

内容	CAN-ID	DATA							
		1	2	3	4	5	6	7	8
ホスト機器から ABH3	0x01ef0102								
ABH3 からホスト機器	0x01ef0201	A パルス積算値				B パルス積算値			

### 2.3 ブロードキャストパケット(0x28)

内容	CAN-ID	DATA							
		1	2	3	4	5	6	7	8
ホスト機器から ABH3	0x00ea0102	0x00ff28							
ABH3 からホスト機器	0x00ff2801	異常フラグ(bit 対応)				警告フラグ(bit 対応)			

### 2.4 ブロードキャストパケット(0x29)

内容	CAN-ID	DATA							
		1	2	3	4	5	6	7	8
ホスト機器から ABH3	0x00ea0102	0x00ff29							
ABH3 からホスト機器	0x00ff2901	制御フラグ(bit 対応)				I O フラグ(bit 対応)			

## 2.5 ブロードキャストパケット(0x2a)

内容	CAN-ID	DATA							
		1	2	3	4	5	6	7	8
ホスト機器から ABH3	0x00ea0102	0x00ff2a							
ABH3 からホスト機器	0x00ff2a01	A/Y 速度指令		B/X 速度指令		A/Y 速度帰還		B/X 速度帰還	

## 2.6 ブロードキャストパケット(0x2b)

内容	CAN-ID	DATA							
		1	2	3	4	5	6	7	8
ホスト機器から ABH3	0x00ea0102	0x00ff2b							
ABH3 からホスト機器	0x00ff2b01	A/Y 電流指令		B/X 電流指令		A 負荷率		B 負荷率	

## 2.7 ブロードキャストパケット(0x2c)

内容	CAN-ID	DATA							
		1	2	3	4	5	6	7	8
ホスト機器から ABH3	0x00ea0102	0x00ff2c							
ABH3 からホスト機器	0x00ff2c01	A パルス積算値				B パルス積算値			

## 2.8 ブロードキャストパケット(0x2d)

内容	CAN-ID	DATA							
		1	2	3	4	5	6	7	8
ホスト機器から ABH3	0x00ea0102	0x00ff2d							
ABH3 からホスト機器	0x00ff2d01	アナログ入力 0		アナログ入力 1		主電源電圧		制御電源電圧	

## 2.9 ブロードキャストパケット(0x2e)

内容	CAN-ID	DATA							
		1	2	3	4	5	6	7	8
ホスト機器から ABH3	0x00ea0102	0x00ff2e							
ABH3 からホスト機器	0x00ff2e01	モニタ 0 データ				モニタ 1 データ			

### 3 データ型

内容	単位	範囲
速度 指令・帰還	0.2[ $\text{min}^{-1}$ ]	-6553.6[ $\text{min}^{-1}$ ]～0[ $\text{min}^{-1}$ ]～6553.4[ $\text{min}^{-1}$ ]
電流 指令・帰還	0.01[%]	-327.68[%]～0.00[%]～327.67[%]
パルス積算値	1[Pulse]	-2147483648[Pulse]～0[Pulse]～2147483647[Pulse]
負荷率	1[%]	0[%]～255[%]
主電源電圧・制御電源電圧	0.1[V]	0.0[V]～6553.5[V]
アナログ入力	0.01[V]	-327.68[V]～0.00[V]～327.67[V]
モニタデータ	—	単精度実数(float)

※複数バイトのデータはリトル・エンディアンとする。

※モニタデータは浮動小数点(float)、その他は整数(short / long)とする。

## 4 関数定義

### 4.1 指令の初期化

関数      `int abh3_can_init(void)`

引数      無し

戻り値    エラー状態 (True : 正常、False : 異常)

内容

指令及び入力のすべてを 0 にし、シングルパケット DP0 を送信する。

シングルパケット DP0 を受信したら戻る。(タイムアウト判定時は異常終了)

A/Y 指令 = 0

B/X 指令 = 0

入力(bit 対応) = 0

DP0 送信／DP0 受信

### 4.2 指令の送信 (軸別)

関数 1      `int abh3_can_cmdAY(short cmd, struct *ptr)`

関数 2      `int abh3_can_cmdBX(short cmd, struct *ptr)`

引数      `cmd` : 指令値

`ptr` : 戻り値の構造体へのポインタ

戻り値      エラー状態 (True : 正常、False : 異常)

内容

関数毎に指令を設定し、シングルパケット DP0 を送信する。

シングルパケット DP0 を受信したら戻り、値を設定する。(タイムアウト判定時は異常終了)

※未指定の指令値と入力は以前のデータを使用する。

関数 1

A/Y 指令 = `cmd`

DP0 送信／DP0 受信

`*ptr` = 受信データ

関数 2

B/X 指令 = `cmd`

DP0 送信／DP0 受信

`*ptr` = 受信データ

### 4.3 指令の送信（同時）

関数            `int abh3_can_cmd(short cmdAY, short cmdBX, struct *ptr)`

引数            `cmdAY` : A/Y 指令値

`cmdBX` : B/X 指令値

`ptr` : 戻り値の構造体へのポインタ

戻り値          エラー状態（True : 正常、False : 異常）

内容

指令を設定し、シングルパケット DP0 を送信する。

シングルパケット DP0 を受信したら戻り、値を設定する。（タイムアウト判定時は異常終了）

※未指定の指令値と入力はい前のデータを使用する。

関数

A/Y 指令 = `cmdAY`

B/X 指令 = `cmdBX`

DP0 送信 / DP0 受信

`*ptr` = 受信データ

### 4.4 入力の送信（一括）

関数            `int abh3_can_inSet(long data, long mask, struct *ptr)`

引数            `data` : データ値

`mask` : マスク値

`ptr` : 戻り値の構造体へのポインタ

戻り値          エラー状態（True : 正常、False : 異常）

内容

マスク値が 1 のデータを入力に設定し、シングルパケット DP0 を送信する。

シングルパケット DP0 を受信したら戻り、値を設定する。（タイムアウト判定時は異常終了）

※未指定の指令値はい前のデータを使用する。

関数

入力 = (入力 & ~`mask`) | (`data` & `mask`)

DP0 送信 / DP0 受信

`*ptr` = 受信データ

## 4.5 入力の送信（ビット）

関数 `int abh3_can_inBitSet(char num, char data, struct *ptr)`

引数 `num` : ビット番号(0~31)

`data` : 設定データ(0~1)

`ptr` : 戻り値の構造体へのポインタ

戻り値 エラー状態 (True : 正常、False : 異常)

内容

ビット番号で指定された入力をデータの値とし、シングルパケット DP0 を送信する。

シングルパケット DP0 を受信したら戻り、値を設定する。(タイムアウト判定時は異常終了)

※未指定の指令値は以前のデータを使用する。

関数

$$\text{入力} = (\text{入力} \ \& \sim(1 \ll \text{num})) \mid (\text{data} \ll \text{num})$$

DP0 送信／DP0 受信

`*ptr` = 受信データ

## 4.6 積算値のリクエスト

関数 `int_abh3_can_reqPulse(struct *ptr)`

引数 `ptr` : 戻り値の構造体へのポインタ

戻り値 エラー状態 (True : 正常、False : 異常)

内容

シングルパケット DP1 を送信する。

シングルパケット DP1 を受信したら戻り、値を設定する。(タイムアウト判定時は異常終了)

関数

DP1 送信／DP1 受信

`*ptr` = 受信データ



#### 4.7 ブロードキャストパケットのリクエスト

関数            `int abh3_can_reqBRD(int num, struct *ptr)`

引数            `num` : 番号(0x00~0xff)

`ptr` : 戻り値の構造体へのポインタ

戻り値          エラー状態 (True : 正常、False : 異常)

内容

引数の番号のブロードキャストリクエストを送信する。

指定したブロードキャストを受信したら戻り、値を設定する。(タイムアウト判定時は異常終了)

関数

BR 送信／BR 受信

`*ptr` = 受信データ

## 4.8 各種変換用関数

種別	変換内容	関数名
速度	float から CAN	short cnvVel2CAN(float vel)
	CAN から float	float cnvCAN2Vel(short vel)
電流	float から CAN	short cnvCur2CAN(float cur)
	CAN から float	float cnvCAN2Cur(short cur)
負荷率	CAN から float	float cnvCAN2Load(short load)
アナログ入力	CAN から float	float cnvCAN2Analog(short analog)
電源電圧	CAN から float	float cnvCAN2Volt(short volt)
実数	float から Pack-float	PACK_FLOAT flt_to_pkflt( float flt )
	Pack-float から float	float pkflt_to_flt( PACK_FLOAT pkflt )
	文字列から Pack-float	int str_to_pkflt( const char *str, PACK_FLOAT *p_pkflt )
	Pack-float から文字列	char *pkflt_to_str( PACK_FLOAT pkflt, char *str )

## 4.9 戻り値の構造体（共用体）

型名	内容
DP0S	A/Y 指令、B/X 指令、入力
DP0R	A 速度帰還、B 速度帰還、Y 速度帰還、X 速度帰還
DP1R	A パルス積算値、B パルス積算値
BR0	異常フラグ、警告フラグ
BR1	制御フラグ、I Oフラグ
BR2	A/Y 速度指令、B/X 速度指令、A/Y 速度帰還、B/X 速度帰還
BR3	A/Y 電流指令、B/X 電流指令、A 負荷率、B 負荷率
BR4	A パルス積算値、B パルス積算値
BR5	アナログ入力 0、アナログ入力 1、主電源電圧、制御電源電圧
BR6	モニタ 0 データ、モニタ 1 データ
BUF	8 バイトデータ