

AJAN
IOコマンド
リファレンス
CAN編

目 次

第1章	はじめに	3
1.1	CAN 通信コマンドについて	3
第2章	機能説明	4
2.1	初期化/終了処理機能.....	4
2.2	メッセージ送信機能.....	6
2.3	メッセージ受信機能.....	6
2.4	動作状態モニタリング	7
2.5	エラー情報取得機能.....	7
2.6	割り込み機能.....	8
2.7	CAN FD 版 CAN 通信.....	11
第3章	リファレンス	12
3.1	コマンド一覧.....	12
3.2	詳細個別説明（初期化/終了）	13
3.3	詳細個別説明（クリア）	16
3.4	詳細個別説明（メッセージ送受信）	17
3.5	詳細個別説明（ステータス）	24
3.6	詳細個別説明（割り込み）	25
3.7	エラーコード一覧.....	29
第4章	サンプルプログラム	30
4.1	サンプルプログラム一覧.....	30
第5章	索引	32
第6章	重要な情報	33

第1章 はじめに

本ドキュメントでは、AJANで利用できるCANのIO制御コマンドの使い方について記載しています。

本ドキュメントでは、説明で表現している表記として下記のように定義します。

- ・ コマンドの書式の説明において、[]内の引数は省略できます。

- ・ 文字の大小について

コマンドは大文字/小文字のどちらでも動作します。

変数名は大文字/小文字も同じものとして扱われます。

ファイルパス/ファイル名は大文字/小文字で区別されます。



本ドキュメント記載の、AJANはIoT用プログラミング言語です。
Interface Linux System上でのみ動作可能です。

1.1 CAN通信コマンドについて

CAN通信コマンドは、フィールドバスの一種であるCAN通信を制御するためのコマンド群です。

コンピュータに標準搭載されているCANポートは、本コマンドを使ってすぐに制御することができますが、弊社インタフェースモジュール等でCANポートを増設した場合、Linuxドライバをインストールする必要があります。

インストールを行い、ドライバの組み込みまで実施する必要があります。

インストール方法、ドライバの組み込み方法については、ドライバ製品のreadme/helpを参照してください。

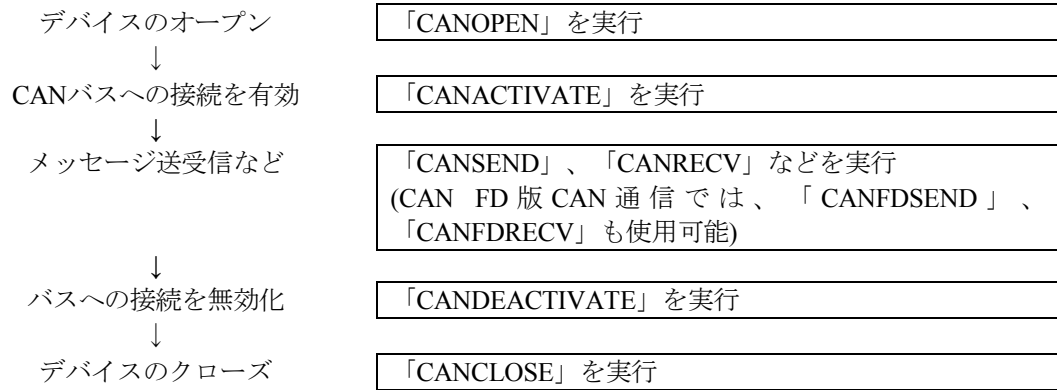
ドライバは、弊社Web siteサイトからダウンロードしてインストールを行ってください。

カテゴリ	型式	リンク先
CAN	GPG-4851	http://www.interface.co.jp/catalog/soft/prdc_soft_all.asp?name=gpg-4851

第2章 機能説明

2.1 初期化/終了処理機能

初期化処理から終了処理までは、以下の流れでコマンドを実行します。



初期化処理は、デバイスを「CANOPEN」で初期化し、「CANACTIVATE」でCANバスへの接続を有効にします。

終了処理は、「CANDEACTIVATE」でバスへの接続を無効化し、「CANCLOSE」でCANインタフェースの終了処理を行います。アプリケーションでは、必ずクローズ処理を行ってから終了してください。

「CANOPEN」で指定するデバイス番号は、/procファイルシステムを参照することで、確認することができます。

デバイス番号は、デバイスノード名のifcanの後の番号になります。

/procファイルシステムで参照できる内容は以下のとおりです。

項目	内容
デバイス名	デバイス名(ifcan1など)を表示します。
デバイス型式	デバイス型式を表示します。
RSW1設定値	RSW1設定値を表示します。
通信速度	現在設定されている通信速度を表示します。 CAN FD製品の場合、2種類の通信速度が表示されます。 ・CANの場合 N：CANの通信速度 ・CANFDの場合 N：CAN FDデータフェーズ以外の通信速度 D：CAN FDデータフェーズの通信速度
送受信通算メッセージ数	現在まで送受信したデータの通算メッセージ数です。 txが送信メッセージ数、rxが受信メッセージ数です。 CAN FD製品の場合は、txfd (CAN FD送信メッセージ数)、rxfd (CAN FD受信メッセージ数)も表示されます。

表示例

通常の CAN 製品の場合

```
# cat /proc/driver/can/cp4851
ifcan1: CPZ/CTP-485220 (P) (bid=0h) CH1 (High-speed) [125000bps] tx:812 rx:4312
ifcan2: CPZ/CTP-485220 (P) (bid=0h) CH2 (High-speed) [125000bps] tx:4312 rx:812
```

上の例では、ifcan1 と ifcan2 のデバイスがあるので、「CANOPEN」に指定できるデバイス番号は、1 と 2 が指定可能。という事になります。

CAN FD 製品の場合

```
表示例
# cp4851 info:1.0
ifcan1: PEX- (H) 485940 (P) (bid=0h) CH1 (High-speed) [N:125000bps / D:1000000bps] tx:10 rx:0 txfd:25 rxfd:18
ifcan2: PEX- (H) 485940 (P) (bid=0h) CH2 (High-speed) [N:125000bps / D:1000000bps] tx:0 rx:10 txfd:18 rxfd:25
ifcan3: PEX- (H) 485940 (P) (bid=0h) CH3 (High-speed) [N:125000bps / D:1000000bps] tx:0 rx:0 txfd:0 rxfd:0
ifcan4: PEX- (H) 485940 (P) (bid=0h) CH4 (High-speed) [N:125000bps / D:1000000bps] tx:0 rx:0 txfd:0 rxfd:0
```

上の例では、ifcan1, ifcan2, ifcan3, ifcan4 のデバイスがあるので、「CANOPEN」に指定できるデバイス番号は、1～4 が指定可能。という事になります。

プログラム例

初期化

```
' CAN ポート 1 のオープン
CANOPEN 1, "BAUDRATE=125000, MODE=NORMAL, TXBSIZE=64, RXBSIZE=64, ERBSIZE=64, ERRLIMIT=96"
' CAN バスへの接続を有効化します。
CANACTIVATE 1
```

終了処理

```
' バスオフ状態 (CAN バスへの接続を無効化) にします。
CANDEACTIVATE 1
' CAN ポート 1 のクローズ
CANCLOSE 1
```

2.2 メッセージ送信機能

「CANSEND」を使用してメッセージの送信を行います。
 (CAN FD版CAN通信の場合は、「CANFDSEND」を使用します)
 送信メッセージはデバイスドライバ内部のバッファに蓄えられ、順次送信が行われます。

●プログラム例

```
DIM CANDATA(7)
CANDATA(0 to 7) = [ &H11; &H22; &H33; &H44; &H55; &H66; &H77; &H88 ]
CANSEND 1, 12, 8, CANDATA ' 1回のCANSENDで、1件のメッセージ送信
```

送信側で送信完了を確認する場合の注意事項

送信バッファはリングバッファになっており、未送信メッセージが送信バッファを1周すると、送信完了メッセージを上書きしてしまいます。

予め送信完了メッセージを全て取得してから、新しい送信メッセージを送信するようにしてください。

この時点で、CANSEND コマンドにより、4つのメッセージを送信バッファに追加するとします。

送信 バッ ファ	
	未送信メッセージ
	未送信メッセージ
	送信完了メッセージ
	送信完了メッセージ
	送信完了メッセージ

未送信メッセージが送信バッファの最後まで挿入され、さらに挿入しようとする...

送信 バッ ファ	未送信メッセージ
	未送信メッセージ
	未送信メッセージ
	未送信メッセージ
	未送信メッセージ
	送信完了メッセージ
	送信完了メッセージ
	送信完了メッセージ

リングバッファなので、次に挿入されるメッセージが、送信完了メッセージを上書きしてしまいます。

送信 バッ ファ	未送信メッセージ
	未送信メッセージ
	未送信メッセージ
	未送信メッセージ
	未送信メッセージ
	送信完了メッセージ
	送信完了メッセージ
	未送信メッセージ

また、送信完了メッセージは、送信バッファサイズを変更するとクリアされますので、送信完了メッセージを全て取り出してから、送信バッファサイズを変更するようにしてください。

2.3 メッセージ受信機能

メッセージ受信は自動的に行われ、デバイスドライバ内部のバッファに蓄えられます。

「CANRECV」はこのバッファから受信メッセージを取り出します。

(CAN FD版CAN通信の場合は、「CANFDRECV」を使用します)

●プログラム例

```
' データの受信
DIM CANDATA(10)
CANDATA = CANRECV(1) ' 1件のメッセージ受信
PRINT "ID      "; CANDATA(0)
PRINT "Time    "; CANDATA(1)
PRINT "Length  "; CANDATA(2)
PRINT "Data(0) "; CANDATA(3)
PRINT "Data(1) "; CANDATA(4)
PRINT "Data(2) "; CANDATA(5)
PRINT "Data(3) "; CANDATA(6)
PRINT "Data(4) "; CANDATA(7)
PRINT "Data(5) "; CANDATA(8)
PRINT "Data(6) "; CANDATA(9)
PRINT "Data(7) "; CANDATA(10)
```

2.4 動作状態モニタリング

「CANSTATUS」で動作状態のステータスをモニタリングします。

※「INCLUDE "IML002CAN.AJN"」で、IML002CAN.AJNをインクルードする必要があります。

●プログラム例

```
' 必要なファイルをインクルードする
INCLUDE "IML002CAN.AJN"
' 変数の宣言
STRUCT TCANSTATUS STATUS
' ステータスを取得
STATUS = CANSTATUS(1)
PRINT "BUSSTATUS="; STATUS.BUSSTATUS
PRINT "TXBCOUNT="; STATUS.TXBCOUNT
PRINT "RXBCOUNT="; STATUS.RXBCOUNT
PRINT "ERBCOUNT="; STATUS.ERBCOUNT
PRINT "TXERRCOUNT="; STATUS.TXERRCOUNT
PRINT "RXERRCOUNT="; STATUS.RXERRCOUNT
PRINT "TXBCOUNTFD="; STATUS.TXBCOUNTFD
PRINT "RXBCOUNTFD="; STATUS.RXBCOUNTFD
```

2.5 エラー情報取得機能

「CANRECVERROR」で、CANバス上で発生したエラーの情報を取得することが出来ます。
 詳細な仕様は「3.4.5 CANRECVERROR」を参照してください。

●プログラム例

```
' データの受信
DIM CANDATA(3)
CANDATA = CANRECVERROR(1)
PRINT "Interrupt="; CANDATA(0)
PRINT "ErrorCode="; CANDATA(1)
PRINT "ArbitrationLost="; CANDATA(2)
PRINT "Time="; CANDATA(3)
```

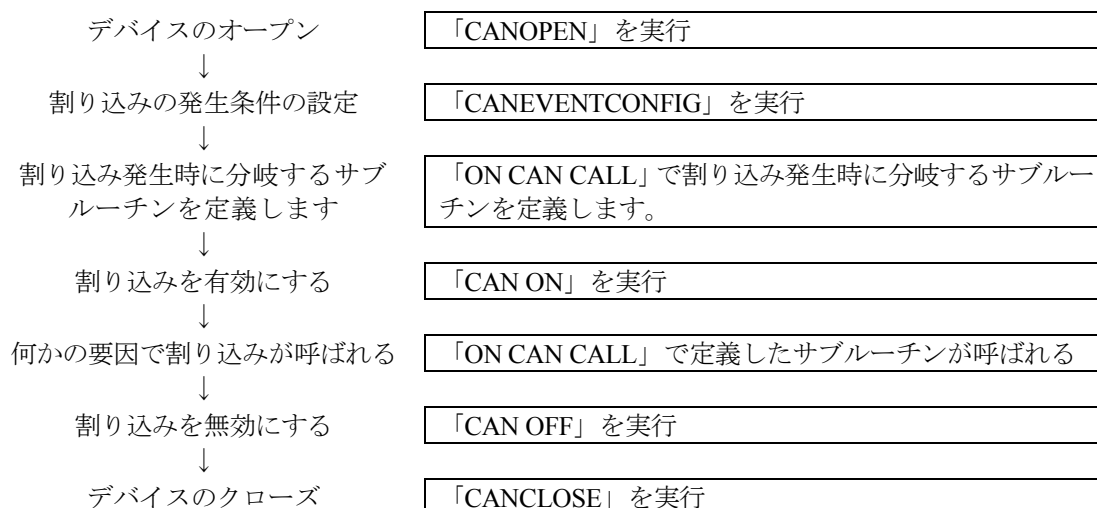
2.6 割り込み機能

デバイスから各割り込みを取得することができます。

取得できる割り込み要因の種別は、以下になります。

内容
エラーバッファがいっぱいになった
受信バッファがいっぱいになった
エラーがリミット値を越えた
送信バッファが空になった
エラーバッファに1つ以上のエラーが蓄えられた
受信バッファに1つ以上のメッセージが蓄えられた
バスステータスが変化した
オーバーランエラーが発生して受信データを取りこぼした

割り込み処理を行うには、以下の流れでコマンドを実行します。



※「INCLUDE "IML002CAN.AJN"」でIML002CAN.AJNをインクルードする必要があります。

※発生した割り込み要因を取得する場合は、「CANEVENT」にて取得します。

●プログラム例

```

' -----
' 割り込み処理
' -----
SUB CANCELLBACK(DN%, EVT%)
    EVENTSTATUS = CANEVENT(DN%)
    PRINT "割り込み発生(要因: "; EVENTSTATUS; ")"
    IF EVENTSTATUS = CANEVTRECV THEN
        LoopFlag = 1
    END IF
END SUB
' -----
' メイン
' -----
' 画面クリア
CLS
' 必要なファイルをインクルードする
INCLUDE "IML002CAN.AJN"
' CAN ポート 1 のオープン
CANOPEN 1, "BAUDRATE=125000"
' CAN バスへの接続を有効化します。
CANACTIVATE 1
' 割り込みの登録
ON CAN 1 CALL CANCELLBACK
' 割り込みの発生条件を設定します
CANEVENTCONFIG 1, "RCV=ON"
' 割り込みを有効にします
CAN ON 1
' 受信バッファに 1 つ以上のメッセージが蓄えられるまで待機
PRINT "受信バッファに 1 つ以上のメッセージが蓄えられるまで待機"
DO WHILE LoopFlag = 0
    SLEEP 0.1 ' ループ内に 0.1 秒のスリープを含めることで、CPU 使用率を低減。
LOOP
' データの受信
DIM CANDATA%(10)
CANDATA% = CANRECV(1)
PRINT "ID      : "; HEX$(CANDATA%(0) AND &H7FFFFFFF)
PRINT "Time    : "; CANDATA%(1)
PRINT "Length  : "; CANDATA%(2)
PRINT "Data(0) : "; CANDATA%(3)
PRINT "Data(1) : "; CANDATA%(4)
PRINT "Data(2) : "; CANDATA%(5)
PRINT "Data(3) : "; CANDATA%(6)
PRINT "Data(4) : "; CANDATA%(7)
PRINT "Data(5) : "; CANDATA%(8)
PRINT "Data(6) : "; CANDATA%(9)
PRINT "Data(7) : "; CANDATA%(10)
' 割り込みの発生条件を無効にします
CANEVENTCONFIG 1, "RCV=OFF"
' 割り込みを無効にします
CAN OFF 1
' バスオフ状態 (CAN バスへの接続を無効化) にします。
CANDEACTIVATE 1
' CAN ポート 1 のクローズ
CANCLOSE 1
' 終了
END

```

送受信時に 29ビットのIDを指定するには、送受信データの配列を、CANDATA% のように整数配列として、明示的に宣言する必要があります。

2.7 CAN FD版CAN通信

CAN FD対応製品で使用すると、CAN FD版CAN通信ができます。これにより より多くのデータを送受信できます。

通常のCAN通信と、CAN FD版CAN通信で、それぞれ使用する命令・関数の相違点を、以下に示します。

CAN FD機能を搭載した CANデバイスに対してオープン(「CANOPEN」)すると、CAN FD通信となります。そうでない CANデバイスに対してオープンすると、CAN通信となります。

機能	CAN通信時 コマンド名	CAN FD版CAN 通信時コマンド名	備考
オープン	CANOPEN	同左	初期化設定パラメータで、設定可能な値に違いがあります。
クローズ	CANCLOSE	同左	
CANバス接続の有効化	CANACTIVATE	同左	
CANバス接続の無効化	CANDEACTIVATE	同左	
クリア	CANCLEAR	同左	
メッセージ送信	CANSEND	CANFDSEND	送信サイズが異なります
メッセージ受信	CANRCV	CANFDRCV	受信サイズが異なります
エラー情報取得	CANRCVERROR	同左	
現在の状態取得	CANSTATUS	同左	一部メンバ値が有効になります。
割り込み条件設定	CANEVENTCONFIG	同左	
割り込み状態取得	CANEVENT	同左	
割り込みルーチン定義	ON CAN CALL	同左	
割り込み有効	CAN ON	同左	
割り込み無効	CAN OFF	同左	
割り込み保留	CAN STOP	同左	

使用するポートが、CAN FD版CAN通信が使用可能かについては、/proc ファイルシステムの通信速度の項を参照すると判別できます。(CAN FD通信時の通信速度を表す「D:」があります)

通常の CAN 製品の場合

```
表示例
# cat /proc/driver/can/cp4851
ifcan1: CPZ/CTP-485220 (P) (bid=0h) CH1 (High-speed) [125000bps] tx:812 rx:4312
ifcan2: CPZ/CTP-485220 (P) (bid=0h) CH2 (High-speed) [125000bps] tx:4312 rx:812
```

CAN FD 製品の場合(CAN FD 版 CAN 通信が可能です)

```
表示例
# cp4851 info:1.0
ifcan1: PEX- (H) 485940 (P) (bid=0h) CH1 (High-speed) [N:125000bps / D:1000000bps] tx:10 rx:0 txfd:25 rxfd:18
ifcan2: PEX- (H) 485940 (P) (bid=0h) CH2 (High-speed) [N:125000bps / D:1000000bps] tx:0 rx:10 txfd:18 rxfd:25
ifcan3: PEX- (H) 485940 (P) (bid=0h) CH3 (High-speed) [N:125000bps / D:1000000bps] tx:0 rx:0 txfd:0 rxfd:0
ifcan4: PEX- (H) 485940 (P) (bid=0h) CH4 (High-speed) [N:125000bps / D:1000000bps] tx:0 rx:0 txfd:0 rxfd:0
```

第3章 リファレンス

3.1 コマンド一覧

コマンド名	機能
●初期化/終了	
CANOPEN	CANポートをオープンします。
CANCLOSE	CANポートをクローズします。
CANACTIVATE	CANバスへの接続を有効化します。
CANDEACTIVATE	バスオフ状態（CANバスへの接続を無効化）にします。
●クリア	
CANCLEAR	デバイスドライバ内に蓄えられている送信、受信メッセージ等を削除します。
●メッセージ送受信	
CANSEND	CANメッセージの送信を行います。
CANFDSEND	CAN FDメッセージの送信を行います。
CANRECV	CANメッセージの受信を行います。
CANFDRECV	CAN FDメッセージの受信を行います。
CANRECVERROR	エラーバッファに蓄えられた受信エラー情報を取り出します。
●ステータス	
CANSTATUS	CANバスの現在の状態を取得します。
●割り込み	
CANEVENTCONFIG	割り込みの発生条件を設定します。
CANEVENT	割り込みの状態を取得します。
ON CAN CALL	割り込み発生時に、呼び出されるサブルーチンを定義します。
CAN ON	割り込みを有効にします。
CAN OFF	割り込みを無効にします。
CAN STOP	割り込みを保留にします。

3.2 詳細個別説明（初期化/終了）

1.CANOPEN

命令			
機 能	CANポートをオープンします。		
書 式	CANOPEN <①デバイス番号>, <②初期化設定>		
パラ メータ	①	<デバイス番号>	数値
	制御するCANデバイスのデバイス番号を設定します。 ※デバイス番号については、「2.1 初期化/終了処理機能」を参照してください。		
	②	<初期化設定>	文字列
	デバイスの初期設定を行います。 「識別子=設定値, 識別子=設定値, …」の文字列形式で渡してください。		
	識別子	設定値	デフォルト
	BAUDRATE	CAN通信時： 33300, 50000, 83300, 95200, 100000, 125000, 250000, 500000, 1000000 CAN FD通信時： 83300, 95200, 100000, 125000, 250000, 500000, 1000000	通信速度 (CAN FD通信を行う場合は、データフェーズ 以外の通信速度を表します) 125000
	BAUDRATEDATA	1000000, 2000000, 5000000	CAN FD製品でのみ設定有効です。 (非CAN FD製品では、この設定は無視されま す) CAN FD通信時における、データフェーズの通 信速度です。 1000000
	MODE	NORMAL	通常の通信モード。 メッセージを受信したらACK を返します。 ○
		MONITOR	モニタモード。 メッセージを受信してもACKを返しません。 ※メッセージの送信を行うことはできません(試行した場合の動作は未定義です)。
	TXBSIZE	64～	送信バッファのサイズをメッセージ単位で 指定します。(「CANSEND」に関係します) 64
	TXBSIZEFD	64～	CAN FD送信バッファのサイズをメッセージ単 位で指定します。(「CANFDSEND」に関係 します) 64
	RXBSIZE	64～	受信バッファのサイズをメッセージ単位で 指定します。(「CANRECV」に関係します) 64
	RXBSIZEFD	64～	CAN FD受信バッファのサイズをメッセージ単 位で指定します。(「CANFDRECV」に関係 します) 64
	ERBSIZE	64～	エラーバッファのサイズをメッセージ単位 で指定します。 64
	ERRLIMIT	CAN通信時： 0～255	送受信の際に発生したエラーが、この件数を 超えるとイベントが発生します。 96
		CAN FD通信： 0～15	
	DEVNAME	デバイス名	デバイス番号に紐付く、任意のデバイス名 (ifcan1など)を指定します。

備 考	<ul style="list-style-type: none"> • CAN FD機能を搭載した CANデバイスに対してオープンすると、CAN FD通信となります。そうでない CANデバイスに対してオープンすると、CAN通信となります。 • CAN FD機能を搭載していない CANデバイスに対して、CAN FDに関する設定(TXBSIZEFDなど)を設定した場合、その設定内容は無効です。 • CAN FD機能を搭載する CANデバイスの場合、BAUDRATE識別子に対して、33300および 50000の通信速度は設定できません。 • BAUDRATE、BAUDRATEDATAについて、選択肢以外の細かい通信速度設定を行いたい場合、Linux用CANドライバGPG-4851の機能をご利用いただくか、弊社サポートにご連絡ください。 • TXBSIZE, TXBSIZEFD, RXBSIZE, RXBSIZEFD, ERBSIZE, ERRLIMIT のサイズや件数の設定できる最大値は環境に依存します。詳しくは、Linux用CANドライバ GPG-4851の説明を参照ください。 • DEVNAME は、ソルコンCDなどの製品で 任意のデバイス名を割り当てる際に用います。 (例：ソルコンCD は、CH1 が ifcan256、CH2 が ifcan257 のようになります)
使用例1	<p>「2.1 初期化/終了処理機能」のプログラム例を参照してください。</p> <p>'CANデバイスを、通信速度 125000bps でオープンします。 CANOPEN 1, "BAUDRATE=125000"</p>
使用例2	<p>'ソルコンCDで、CH1 が ifcan256 の際に、DEVNAMEを使って割り当てる例です。 CANOPEN 1, "BAUDRATE=125000,DEVNAME=ifcan256"</p>

2.CANCLOSE

命令			
機能	CANポートをクローズします。		
書式	CANCLOSE <①デバイス番号>		
パラメータ	①	<デバイス番号>	数値
	「CANOPEN」でオープンしたデバイス番号を指定します。		
使用例	「2.1 初期化/終了処理機能」のプログラム例を参照してください。 'CANデバイスをオープンした後、クローズします。 CANOPEN 1, "BAUDRATE=125000" CANCLOSE 1		

3.CANACTIVATE

命令			
機能	CANバスへの接続を有効化します。 データの送受信が行える状態になります。		
書式	CANACTIVATE <①デバイス番号>		
パラメータ	①	<デバイス番号>	数値
	「CANOPEN」でオープンしたデバイス番号を指定します。		
使用例	「2.1 初期化/終了処理機能」のプログラム例を参照してください。 'CANデバイスをオープンした後、接続を有効化します CANOPEN 1, "BAUDRATE=125000" CANACTIVATE 1 ' 接続を有効化した後、送受信できます。		

4.CANDEACTIVATE

命令			
機能	バスオフ状態（CANバスへの接続を無効化）にします。 メッセージの送受信が行えない状態になります。		
書式	CANDEACTIVATE <①デバイス番号>		
パラメータ	①	<デバイス番号>	数値
	「CANOPEN」でオープンしたデバイス番号を指定します。		
使用例	「2.1 初期化/終了処理機能」のプログラム例を参照してください。 'CANデバイスをオープンした後、接続を有効化し、その後、無効化します。 CANOPEN 1, "BAUDRATE=125000" CANACTIVATE 1 ' 接続を有効化した後、送受信できます。 CANDEACTIVATE 1 ' 接続を無効化した後、送受信できません。		

3.3 詳細個別説明（クリア）

1. CANCELAR

命令																																																
機 能	デバイスドライバ内に蓄えられている送信、受信メッセージ等を削除します。																																															
書 式	CANCELAR <①デバイス番号>, <②対象>																																															
パラ メータ	①	<デバイス番号>	数値																																													
	「CANOPEN」でオープンしたデバイス番号を指定します。																																															
	②	<対象>	文字列																																													
	クリアするバッファの種類を指定します。 「識別子=設定値, 識別子=設定値, ...」の文字列形式で渡してください。																																															
	<table border="1"> <thead> <tr> <th>識別子</th><th>設定値</th><th>内容</th><th>デフォルト</th></tr> </thead> <tbody> <tr> <td rowspan="2">TXB</td><td>OFF</td><td>送信バッファをクリアしない。</td><td>○</td></tr> <tr> <td>ON</td><td>送信バッファをクリアする。</td><td></td></tr> <tr> <td rowspan="2">RXB</td><td>OFF</td><td>受信バッファをクリアしない。</td><td>○</td></tr> <tr> <td>ON</td><td>受信バッファをクリアする。</td><td></td></tr> <tr> <td rowspan="2">TXERR</td><td>OFF</td><td>送信エラーカウンタをクリアしない。</td><td>○</td></tr> <tr> <td>ON</td><td>送信エラーカウンタをクリアする。</td><td></td></tr> <tr> <td rowspan="2">RXERR</td><td>OFF</td><td>受信エラーカウンタをクリアしない。</td><td>○</td></tr> <tr> <td>ON</td><td>受信エラーカウンタをクリアする。</td><td></td></tr> <tr> <td rowspan="2">ERB</td><td>OFF</td><td>エラーバッファをクリアしない。</td><td>○</td></tr> <tr> <td>ON</td><td>エラーバッファをクリアする。</td><td></td></tr> <tr> <td rowspan="2">TIMERCOUNTER</td><td>OFF</td><td>内部カウンタをクリアしない。</td><td>○</td></tr> <tr> <td>ON</td><td>内部カウンタをクリアする。</td><td></td></tr> </tbody> </table>			識別子	設定値	内容	デフォルト	TXB	OFF	送信バッファをクリアしない。	○	ON	送信バッファをクリアする。		RXB	OFF	受信バッファをクリアしない。	○	ON	受信バッファをクリアする。		TXERR	OFF	送信エラーカウンタをクリアしない。	○	ON	送信エラーカウンタをクリアする。		RXERR	OFF	受信エラーカウンタをクリアしない。	○	ON	受信エラーカウンタをクリアする。		ERB	OFF	エラーバッファをクリアしない。	○	ON	エラーバッファをクリアする。		TIMERCOUNTER	OFF	内部カウンタをクリアしない。	○	ON	内部カウンタをクリアする。
識別子	設定値	内容	デフォルト																																													
TXB	OFF	送信バッファをクリアしない。	○																																													
	ON	送信バッファをクリアする。																																														
RXB	OFF	受信バッファをクリアしない。	○																																													
	ON	受信バッファをクリアする。																																														
TXERR	OFF	送信エラーカウンタをクリアしない。	○																																													
	ON	送信エラーカウンタをクリアする。																																														
RXERR	OFF	受信エラーカウンタをクリアしない。	○																																													
	ON	受信エラーカウンタをクリアする。																																														
ERB	OFF	エラーバッファをクリアしない。	○																																													
	ON	エラーバッファをクリアする。																																														
TIMERCOUNTER	OFF	内部カウンタをクリアしない。	○																																													
	ON	内部カウンタをクリアする。																																														
備 考	TXERR, RXERR を指定するとき、バスオフ状態でのみ使用可能です。 使用時にはCANDEACTIVATE関数で、バスオフ状態にする必要があります。 バスオフ状態になっていない場合、エラーが返され、全てのクリアは行われません																																															
使用例	' 送信バッファ(TXB)、受信バッファ(RXB)、内部カウンタ(TIMERCOUNTER) をクリアします。 CANCELAR 1, "TXB=ON, RXB=ON, TIMERCOUNTER=ON"																																															

3.4 詳細個別説明（メッセージ送受信）

1. CANSEND

命令	
機能	CANメッセージの送信を行います。
書式	CANSEND <①デバイス番号>, <②ID>, <③送信サイズ> [, <④送信データ>]
パラメータ	①

2. CANFDSEND

命令									
機能	CAN FDメッセージの送信を行います。								
書式	CANFDSEND <①デバイス番号>, <②ID>, <③送信サイズ> [, <④送信データ>]								
パラメータ	①	<デバイス番号>					数値		
	「CANOPEN」でオープンしたデバイス番号を指定します。								
	②	<ID>					数値		
	送信メッセージに付加するIDを指定します。								
	11ビットと29ビットのIDの区別は以下の値をORすることで行うことができます。 (指定しない場合は11ビットIDとして上位を無視して下位11ビットのみがIDとして設定されます。)								
	値		内容						
	&H80000000&		IDは29ビットIDです。						
	11ビットIDは、以下のビットパターンです。								
	bit31		bit30	...		bit11	bit10	bit9	bit8
	0				無効		ID. 10	ID. 9	ID. 8
bit7		bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ID. 7		ID. 6	ID. 5	ID. 4	ID. 3	ID. 2	ID. 1	ID. 0	
29ビットIDは、以下のビットパターンです。									
bit31		bit30	bit29	bit28	bit27	bit26	bit25	bit24	
1				ID.28	ID.27	ID.26	ID.25	ID.24	
bit23		bit22	bit21	bit20	bit19	bit18	bit17	bit16	
ID.23		ID.22	ID.21	ID.20	ID.19	ID.18	ID.17	ID.16	
bit15		bit14	bit13	bit12	bit11	bit10	bit9	bit8	
ID.15		ID.14	ID.13	ID.12	ID.11	ID.10	ID.9	ID.8	
bit7		bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ID.7		ID.6	ID.5	ID.4	ID.3	ID.2	ID.1	ID.0	
備考	③	<送信サイズ>					数値		
	送信するデータのサイズを0～8, 12, 16, 20, 24, 32, 48, 64の範囲で指定します。								
	④	<送信データ>					配列		
	送信するデータを指定します。 データは配列に格納します。 配列の要素数は、送信サイズ以上を与えてください。 送信サイズが0の場合は省略できます。								
備考	・この命令は、CAN FD機能を搭載したCANデバイスでのみ 使用可能です。								
使用例	「2.2 メッセージ送信機能」のプログラム例を参照してください。 ' 拡張フレームを使ってIDに16進数の&H8C0(10進数では2240)を指定する例 ID_VAR%= &H8C0 OR &H80000000 DIM CANDATA(7) CANDATA(0 to 7)= [&H11;&H22;&H33;&H44;&H55;&H66;&H77;&H88] CANFDSEND 1, ID_VAR%, 8, CANDATA ' 送信サイズ：8のデータを送信します。								

3.CANRECV

関数																																							
機 能	CANメッセージの受信を行います。																																						
書 式	<(戻り値)受信メッセージ>=CANRECV(<①デバイス番号>)																																						
戻り値	戻り値	<受信メッセージ>	配列																																				
	受信したメッセージは、11要素の配列内に次のように格納されます。																																						
	<table><tr><th>添字</th><th>内容</th><th>説明</th></tr><tr><td>(0)</td><td>ID</td><td>受信メッセージに付加されているID。</td></tr><tr><td>(1)</td><td>Time</td><td>メッセージを受信した時間（μs） (CANOPEN実行後からの経過時間)</td></tr><tr><td>(2)</td><td>Length</td><td>受信メッセージのサイズ</td></tr><tr><td>(3)</td><td>Data(0)</td><td>受信メッセージ1バイト目</td></tr><tr><td>(4)</td><td>Data(1)</td><td>受信メッセージ2バイト目</td></tr><tr><td>(5)</td><td>Data(2)</td><td>受信メッセージ3バイト目</td></tr><tr><td>(6)</td><td>Data(3)</td><td>受信メッセージ4バイト目</td></tr><tr><td>(7)</td><td>Data(4)</td><td>受信メッセージ5バイト目</td></tr><tr><td>(8)</td><td>Data(5)</td><td>受信メッセージ6バイト目</td></tr><tr><td>(9)</td><td>Data(6)</td><td>受信メッセージ7バイト目</td></tr><tr><td>(10)</td><td>Data(7)</td><td>受信メッセージ8バイト目</td></tr></table>			添字	内容	説明	(0)	ID	受信メッセージに付加されているID。	(1)	Time	メッセージを受信した時間（μs） (CANOPEN実行後からの経過時間)	(2)	Length	受信メッセージのサイズ	(3)	Data(0)	受信メッセージ1バイト目	(4)	Data(1)	受信メッセージ2バイト目	(5)	Data(2)	受信メッセージ3バイト目	(6)	Data(3)	受信メッセージ4バイト目	(7)	Data(4)	受信メッセージ5バイト目	(8)	Data(5)	受信メッセージ6バイト目	(9)	Data(6)	受信メッセージ7バイト目	(10)	Data(7)	受信メッセージ8バイト目
	添字	内容	説明																																				
	(0)	ID	受信メッセージに付加されているID。																																				
(1)	Time	メッセージを受信した時間（μs） (CANOPEN実行後からの経過時間)																																					
(2)	Length	受信メッセージのサイズ																																					
(3)	Data(0)	受信メッセージ1バイト目																																					
(4)	Data(1)	受信メッセージ2バイト目																																					
(5)	Data(2)	受信メッセージ3バイト目																																					
(6)	Data(3)	受信メッセージ4バイト目																																					
(7)	Data(4)	受信メッセージ5バイト目																																					
(8)	Data(5)	受信メッセージ6バイト目																																					
(9)	Data(6)	受信メッセージ7バイト目																																					
(10)	Data(7)	受信メッセージ8バイト目																																					
パラメータ	①	<デバイス番号>	数値																																				
	「CANOPEN」でオープンしたデバイス番号を指定します。																																						
備 考	・この関数は、CAN FD機能を搭載したCANデバイス、通常のCANデバイス、共に使用可能です。																																						
使用例	「2.3 メッセージ受信機能」のプログラム例を参照してください。 DIM CANDATA(10) 'CANメッセージを受信します。 CANDATA=CANRECV(1) '受信データを表示します FOR I=0 TO CANDATA(2)-1 PRINT CANDATA(I+3) NEXT I																																						

4.CANFDRECV

関数																																	
機 能	CAN FDメッセージの受信を行います。																																
書 式	<(戻り値)受信メッセージ> = CANFDRECV(<①デバイス番号>)																																
戻り値	戻り値	<受信メッセージ>	配列																														
	受信したメッセージは、67要素の配列内に次のように格納されます。																																
	<table><tr><th>添字</th><th>内容</th><th>説明</th></tr><tr><td>(0)</td><td>ID</td><td>受信メッセージに付加されているID。</td></tr><tr><td>(1)</td><td>Time</td><td>メッセージを受信した時間 (μs) (CANOPEN実行後からの経過時間)</td></tr><tr><td>(2)</td><td>Length</td><td>受信メッセージのサイズ</td></tr><tr><td>(3)</td><td>Data(0)</td><td>受信メッセージ1バイト目</td></tr><tr><td>(4)</td><td>Data(1)</td><td>受信メッセージ2バイト目</td></tr><tr><td>(5)</td><td>Data(2)</td><td>受信メッセージ3バイト目</td></tr><tr><td>...</td><td>...</td><td>...</td></tr><tr><td>(65)</td><td>Data(62)</td><td>受信メッセージ63バイト目</td></tr><tr><td>(66)</td><td>Data(63)</td><td>受信メッセージ64バイト目</td></tr></table>			添字	内容	説明	(0)	ID	受信メッセージに付加されているID。	(1)	Time	メッセージを受信した時間 (μs) (CANOPEN実行後からの経過時間)	(2)	Length	受信メッセージのサイズ	(3)	Data(0)	受信メッセージ1バイト目	(4)	Data(1)	受信メッセージ2バイト目	(5)	Data(2)	受信メッセージ3バイト目	(65)	Data(62)	受信メッセージ63バイト目	(66)	Data(63)	受信メッセージ64バイト目
	添字	内容	説明																														
(0)	ID	受信メッセージに付加されているID。																															
(1)	Time	メッセージを受信した時間 (μs) (CANOPEN実行後からの経過時間)																															
(2)	Length	受信メッセージのサイズ																															
(3)	Data(0)	受信メッセージ1バイト目																															
(4)	Data(1)	受信メッセージ2バイト目																															
(5)	Data(2)	受信メッセージ3バイト目																															
...																															
(65)	Data(62)	受信メッセージ63バイト目																															
(66)	Data(63)	受信メッセージ64バイト目																															
パラメータ	①	<デバイス番号>	数値																														
	「CANOPEN」でオープンしたデバイス番号を指定します。																																
備 考	・ この関数は、CAN FD機能を搭載したCANデバイスでのみ 使用可能です。																																
使用例	「2.3 メッセージ受信機能」のプログラム例を参照してください。 DIM CANDATA(67) ' CANメッセージを受信します。 CANDATA = CANFDRECV(1) ' 受信データを表示します FOR I=0 TO CANDATA(2)-1 PRINT CANDATA(I+3) NEXT I																																

5.CANRECVERROR

関数		
機 能	エラーバッファに蓄えられた受信エラー情報を取り出します。	
書 式	<(戻り値)受信エラー情報> = CANRECVERROR(<①デバイス番号>)	
戻り値	戻り値	<受信エラー情報> 配列
	受信したエラー情報は、4要素の配列内に次のように格納されます。 各要素の詳細は、「<受信エラー情報の詳細説明>」を参照ください。	
	添字	内容 説明
	(0)	Interrupt エラー割り込み発生時の発生要因
	(1)	ErrorCode CANバス上のエラーが発生した場合の詳細な内容
	(2)	ArbitrationLost アービトレーション・ロスト
	(3)	Time エラーが発生した時刻 (μs)
パラメータ	①	<デバイス番号> 数値
	「CANOPEN」でオープンしたデバイス番号を指定します。	
備 考	エラー情報があるかどうかは、「CANSTATUS」でエラーバッファの状態を見ることで確認できます。	
使用例	「2.5 エラー情報取得機能」のプログラム例を参照してください。 DIM INFO(3) ' エラー情報を取得します。 INFO = CANRECVERROR(1) PRINT "エラー情報 : "; INFO	

<受信エラー情報の詳細説明>

・Interrupt の詳細説明

エラー割り込み発生時の発生要因がセットされます。

複数の要因が同時に成立している場合もあります。

値	内容
&H00000080	CANバス上のエラーを検出しました。 詳細はErrorCodeでご確認ください。
&H00000040	CANバス上でアービトレーションロストを検出しました。 詳細はArbitrationLostでご確認ください。
&H00000020	エラーバシブ状態となりました。(送信か受信のエラーカウンタが127を越えました)
&H00000008	オーバーランエラーが発生し、受信メッセージが失われました。
&H00000004	送信か受信のエラーカウンタが、「CANOPEN」の ERRLIMIT で設定したリミット値を越えました。

・ErrorCode の詳細説明

CANバス上のエラーでない場合は0が返されます。(CANコントローラ SJA1000のECCレジスタに相当します)

bit31～8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
予約	ERRC1	ERRC0	DIR	SEG4	SEG3	SEG2	SEG1	SEG0

各ビットの意味

ERRC1	ERRC0	エラーの種類
0	0	ビットエラー
0	1	フォームエラー
1	0	ビットスタッフエラー
1	1	その他の種類のエラー

DIR	エラーの発生した方向
0	送信時に発生したエラー

IO制御コマンドリファレンス

1	受信時に発生したエラー
---	-------------

SEG4	SEG3	SEG2	SEG1	SEG0	エラーの発生箇所
0	0	0	1	1	SOF (スタートオブフレーム)
0	0	0	1	0	ID.28～ID.21の間
0	0	1	1	0	ID.20～ID.18の間
0	0	1	0	0	SRTR (標準フォーマット時はRTR)
0	0	1	0	1	IDE
0	0	1	1	1	ID.17～ID.13の間
0	1	1	1	1	ID.12～ID.5の間
0	1	1	1	0	ID.4～ID.0の間
0	1	1	0	0	RTR
0	1	1	0	1	R1 (予約ビット)
0	1	0	0	1	R0 (予約ビット)
0	1	0	1	1	DLC (データ長コード)
0	1	0	1	0	データフィールド
0	1	0	0	0	CRCシーケンス
1	1	0	0	0	CRCデリミタ
1	1	0	0	1	ACKスロット
1	1	0	1	1	ACKデリミタ
1	1	0	1	0	EOF (エンドオブフレーム)
1	0	0	1	0	インターミッション
1	0	0	0	1	アクティブエラーフラグ
1	0	1	1	0	パッシブエラーフラグ
1	0	0	1	1	重ね合わせエラーフラグ
1	0	1	1	1	エラーデリミタ
1	1	1	0	0	オーバーロードフラグ

※CAN FD版CAN製品ではビット割り当てが異なります。以下を参照してください。

bit31～8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
予約	ERRC2	ERRC1	ERRC0	予約	予約	予約	予約	予約

各ビットの意味

ERRC2	ERRC1	ERRC0	エラーの種類
0	0	0	エラー無し
0	0	1	ビットエラー
0	1	0	フォームエラー
0	1	1	スタッフエラー
1	0	0	ACKエラー
1	0	1	CRCエラー
1	1	0	その他のエラー
1	1	1	予約

・ ArbitrationLost の詳細説明

CANバス上で、送信データの衝突が発生し、その際の調停で負けた場合、どのビットで調停に負けたか(アービトレーション・ロスト) を記録するレジスタです。

Interruptのアービトレーションロストを検出してない場合は0が返されます。(CANコントローラ SJA1000のALCレジスタに相当します)

bit31～5	bit4	bit3	bit2	bit1	bit0
予約	BITNO4	BITNO3	BITNO2	BITNO1	BITNO0

各ビットの意味は下記ようになります。

BITNO4	BITNO3	BITNO2	BITNO1	BITNO0	アービトレーションロスト位置
0	0	0	0	0	IDの1ビット目 (※1)
0	0	0	0	1	IDの2ビット目
0	0	0	1	0	IDの3ビット目
0	0	0	1	1	IDの4ビット目
0	0	1	0	0	IDの5ビット目
0	0	1	0	1	IDの6ビット目
0	0	1	1	0	IDの7ビット目
0	0	1	1	1	IDの8ビット目

0	1	0	0	0	IDの9ビット目
0	1	0	0	1	IDの10ビット目
0	1	0	1	0	IDの11ビット目
0	1	0	1	1	SRTRビット (※2)(※4)
0	1	1	0	0	IDEビット
0	1	1	0	1	IDの12ビット目 (※3)
0	1	1	1	0	IDの13ビット目 (※3)
0	1	1	1	1	IDの14ビット目 (※3)
1	0	0	0	0	IDの15ビット目 (※3)
1	0	0	0	1	IDの16ビット目 (※3)
1	0	0	1	0	IDの17ビット目 (※3)
1	0	0	1	1	IDの18ビット目 (※3)
1	0	1	0	0	IDの19ビット目 (※3)
1	0	1	0	1	IDの20ビット目 (※3)
1	0	1	1	0	IDの21ビット目 (※3)
1	0	1	1	1	IDの22ビット目 (※3)
1	1	0	0	0	IDの23ビット目 (※3)
1	1	0	0	1	IDの24ビット目 (※3)
1	1	0	1	0	IDの25ビット目 (※3)
1	1	0	1	1	IDの26ビット目 (※3)
1	1	1	0	0	IDの27ビット目 (※3)
1	1	1	0	1	IDの28ビット目 (※3)
1	1	1	1	0	IDの29ビット目 (※3)
1	1	1	1	1	RTRビット (※3)(※4)

※1. IDの1ビット目はID.28、2ビット目はID.27、…、となります。

※2. 標準フォーマットの場合、RTRビット。

※3. 拡張フォーマットの場合のみ有効。

※4. CAN FD の場合はRRSビット

• Timeの詳細説明

実際の時刻ではなく、「CANOPEN」の実行後からの経過時間になります。

受信メッセージにも同じ時刻が記録されますので、受信メッセージとエラー発生の時間的関係を結びつけることができます。

「CANCLEAR」で内部カウンタのクリアを指定することで、内部の経過時間を0クリアすることが出来ます。

単位は μs ですが、実際に返される値はスタンダード版では1000単位の値(ms)が返されます。高機能版、バスマスタ版CANデバイスでは、10単位の値が返されます。

また、この時間は、スタンダード版CANデバイスでは、CANコントローラから割り込みが発生し、デバイスドライバの割り込み処理内でCANメッセージを取り出した時の時間ですので、正確性には欠けます。

また、連続でCANメッセージを受信した場合、2つのメッセージの受信時刻が同じになることがあります。

3.5 詳細個別説明（ステータス）

1. CANSTATUS

関数																																						
機能	CANバスの現在の状態を取得します。																																					
書式	<(戻り値)CANバス状態> = CANSTATUS(<①デバイス番号>)																																					
戻り値	戻り値	<CANバス状態>	TCANSTATUS構造体																																			
	<p>CANバスの状態を取得します。 詳細は下記の通りになります。</p> <p>構造体名 TCANSTATUS</p> <pre> DEFINE STRUCT TCANSTATUS BUSSTATUS TXBCOUNT RXBCOUNT ERBCOUNT TXERRCOUNT RXERRCOUNT TXBCOUNTFD RXBCOUNTFD END STRUCT </pre> <table border="1"> <thead> <tr> <th>構造体メンバ変数</th><th>内容</th><th>型</th><th>値</th></tr> </thead> <tbody> <tr> <td>BUSSTATUS</td><td>CANバスの状態</td><td>数値</td><td>1：バスは正常に動作しています。(エラーアクティブ状態です。) 2：送信か受信のエラーカウンタが 127 を越えました。(エラーパッシブ状態です) 4：バスオフ状態です。</td></tr> <tr> <td>TXBCOUNT</td><td>送信バッファに蓄えられているCANメッセージの数</td><td>数値</td><td>0 ～ 最大値</td></tr> <tr> <td>RXBCOUNT</td><td>受信バッファに蓄えられているCANメッセージの数</td><td>数値</td><td>0 ～ 最大値</td></tr> <tr> <td>ERBCOUNT</td><td>エラーバッファに蓄えられているエラー情報の数</td><td>数値</td><td>0 ～ 最大値</td></tr> <tr> <td>TXERRCOUNT</td><td>送信エラーカウンタの値</td><td>数値</td><td>0 ～ 255</td></tr> <tr> <td>RXERRCOUNT</td><td>受信エラーカウンタの値</td><td>数値</td><td>0 ～ 255</td></tr> <tr> <td>TXBCOUNTFD</td><td>送信バッファに蓄えられているCAN FDメッセージの数</td><td>数値</td><td>0 ～ 最大値</td></tr> <tr> <td>RXBCOUNTFD</td><td>受信バッファに蓄えられているCAN FDメッセージの数</td><td>数値</td><td>0 ～ 最大値</td></tr> </tbody> </table> <p>※最大値は CANOPEN コマンドで設定した値です。</p>			構造体メンバ変数	内容	型	値	BUSSTATUS	CANバスの状態	数値	1：バスは正常に動作しています。(エラーアクティブ状態です。) 2：送信か受信のエラーカウンタが 127 を越えました。(エラーパッシブ状態です) 4：バスオフ状態です。	TXBCOUNT	送信バッファに蓄えられているCANメッセージの数	数値	0 ～ 最大値	RXBCOUNT	受信バッファに蓄えられているCANメッセージの数	数値	0 ～ 最大値	ERBCOUNT	エラーバッファに蓄えられているエラー情報の数	数値	0 ～ 最大値	TXERRCOUNT	送信エラーカウンタの値	数値	0 ～ 255	RXERRCOUNT	受信エラーカウンタの値	数値	0 ～ 255	TXBCOUNTFD	送信バッファに蓄えられているCAN FDメッセージの数	数値	0 ～ 最大値	RXBCOUNTFD	受信バッファに蓄えられているCAN FDメッセージの数	数値
構造体メンバ変数	内容	型	値																																			
BUSSTATUS	CANバスの状態	数値	1：バスは正常に動作しています。(エラーアクティブ状態です。) 2：送信か受信のエラーカウンタが 127 を越えました。(エラーパッシブ状態です) 4：バスオフ状態です。																																			
TXBCOUNT	送信バッファに蓄えられているCANメッセージの数	数値	0 ～ 最大値																																			
RXBCOUNT	受信バッファに蓄えられているCANメッセージの数	数値	0 ～ 最大値																																			
ERBCOUNT	エラーバッファに蓄えられているエラー情報の数	数値	0 ～ 最大値																																			
TXERRCOUNT	送信エラーカウンタの値	数値	0 ～ 255																																			
RXERRCOUNT	受信エラーカウンタの値	数値	0 ～ 255																																			
TXBCOUNTFD	送信バッファに蓄えられているCAN FDメッセージの数	数値	0 ～ 最大値																																			
RXBCOUNTFD	受信バッファに蓄えられているCAN FDメッセージの数	数値	0 ～ 最大値																																			
パラメータ	①	<デバイス番号>	数値																																			
	「CANOPEN」でオープンしたデバイス番号を指定します。																																					
備考	本コマンドを使用する際は、INCLUDEコマンドでIML002CAN.AJNをインクルードしてください。																																					
使用例	<p>「2.4 動作状態モニタリング」のプログラム例を参照してください。</p> <pre> ' 構造体の定義の為に、プログラム先頭に必要 INCLUDE "IML002CAN.AJN" ' 変数の宣言 STRUCT TCANSTATUS STATUS ' CANバスの動作状態を取得します STATUS = CANSTATUS(1) PRINT STATUS </pre>																																					

3.6 詳細個別説明（割り込み）

1. CANEVENTCONFIG

命令

機能	割り込みの発生条件を設定します。																																																														
書式	CANEVENTCONFIG <①デバイス番号> , <②割り込み定義>																																																														
パラメータ	①	<デバイス番号>	数値																																																												
	「CANOPEN」でオープンしたデバイス番号を指定します。																																																														
	②	<割り込み定義>	文字列																																																												
	割り込みの定義を設定します。																																																														
	<table><tr><th>種類</th><th>説明</th></tr><tr><td>ERB</td><td>エラーバッファがいっぱいになった</td></tr><tr><td>RXB</td><td>受信バッファがいっぱいになった</td></tr><tr><td>LIMIT</td><td>エラーがリミット値を越えた</td></tr><tr><td>TXB</td><td>送信バッファが空になった</td></tr><tr><td>ERROR</td><td>エラーバッファに1つ以上のエラーが蓄えられた</td></tr><tr><td>RECV</td><td>受信バッファに1つ以上のメッセージが蓄えられた</td></tr><tr><td>BUSSTATUS</td><td>バスステータスが変化した</td></tr><tr><td>OVERRUN</td><td>オーバーランエラーが発生し受信データを取りこぼした</td></tr></table>			種類	説明	ERB	エラーバッファがいっぱいになった	RXB	受信バッファがいっぱいになった	LIMIT	エラーがリミット値を越えた	TXB	送信バッファが空になった	ERROR	エラーバッファに1つ以上のエラーが蓄えられた	RECV	受信バッファに1つ以上のメッセージが蓄えられた	BUSSTATUS	バスステータスが変化した	OVERRUN	オーバーランエラーが発生し受信データを取りこぼした																																										
種類	説明																																																														
ERB	エラーバッファがいっぱいになった																																																														
RXB	受信バッファがいっぱいになった																																																														
LIMIT	エラーがリミット値を越えた																																																														
TXB	送信バッファが空になった																																																														
ERROR	エラーバッファに1つ以上のエラーが蓄えられた																																																														
RECV	受信バッファに1つ以上のメッセージが蓄えられた																																																														
BUSSTATUS	バスステータスが変化した																																																														
OVERRUN	オーバーランエラーが発生し受信データを取りこぼした																																																														
	「識別子=設定値, 識別子=設定値, …」の文字列形式で渡してください。																																																														
	例: "ERB=ON, RXB=ON"																																																														
	■対応表																																																														
	<table><tr><th>識別子</th><th>設定値</th><th>内容</th><th>デフォルト</th></tr><tr><td rowspan="2">ERB</td><td>OFF</td><td>無効</td><td>○</td></tr><tr><td>ON</td><td>有効</td><td></td></tr><tr><td rowspan="2">RXB</td><td>OFF</td><td>無効</td><td>○</td></tr><tr><td>ON</td><td>有効</td><td></td></tr><tr><td rowspan="2">LIMIT</td><td>OFF</td><td>無効</td><td>○</td></tr><tr><td>ON</td><td>有効</td><td></td></tr><tr><td rowspan="2">TXB</td><td>OFF</td><td>無効</td><td>○</td></tr><tr><td>ON</td><td>有効</td><td></td></tr><tr><td rowspan="2">ERROR</td><td>OFF</td><td>無効</td><td>○</td></tr><tr><td>ON</td><td>有効</td><td></td></tr><tr><td rowspan="2">RECV</td><td>OFF</td><td>無効</td><td>○</td></tr><tr><td>ON</td><td>有効</td><td></td></tr><tr><td rowspan="2">BUSSTATUS</td><td>OFF</td><td>無効</td><td>○</td></tr><tr><td>ON</td><td>有効</td><td></td></tr><tr><td rowspan="2">OVERRUN</td><td>OFF</td><td>無効</td><td>○</td></tr><tr><td>ON</td><td>有効</td><td></td></tr></table>			識別子	設定値	内容	デフォルト	ERB	OFF	無効	○	ON	有効		RXB	OFF	無効	○	ON	有効		LIMIT	OFF	無効	○	ON	有効		TXB	OFF	無効	○	ON	有効		ERROR	OFF	無効	○	ON	有効		RECV	OFF	無効	○	ON	有効		BUSSTATUS	OFF	無効	○	ON	有効		OVERRUN	OFF	無効	○	ON	有効	
識別子	設定値	内容	デフォルト																																																												
ERB	OFF	無効	○																																																												
	ON	有効																																																													
RXB	OFF	無効	○																																																												
	ON	有効																																																													
LIMIT	OFF	無効	○																																																												
	ON	有効																																																													
TXB	OFF	無効	○																																																												
	ON	有効																																																													
ERROR	OFF	無効	○																																																												
	ON	有効																																																													
RECV	OFF	無効	○																																																												
	ON	有効																																																													
BUSSTATUS	OFF	無効	○																																																												
	ON	有効																																																													
OVERRUN	OFF	無効	○																																																												
	ON	有効																																																													
使用例	「2.6 割り込み機能」のプログラム例を参照してください。 ' 受信バッファにメッセージが蓄えられると、割り込みが入るように指定します。 CANEVENTCONFIG 1, "RECV=ON"																																																														

2.CANEVENT

関数			
機 能	割り込みの状態を取得します。		
書 式	<(戻り値)割り込み状態> = CANEVENT(<①デバイス番号>)		
戻り値	戻り値	<割り込み状態>	数値
	割り込み状態を取得します。		
	■ 対応表		
	識別子	設定値	内容
	CANEVTERB	1	エラーバッファがいっぱいになった
CANEVTRXB	2	受信バッファがいっぱいになった	
CANEVTLIMIT	3	エラーがリミット値を越えた	
CANEVTTXB	4	送信バッファが空になった	
CANEVTERORR	5	エラーバッファに 1 つ以上のエラーが蓄えられた	
CANEVTRECV	6	受信バッファに 1 つ以上のメッセージが蓄えられた	
CANEVTBUSSTATUS	7	バスステータスが変化した	
CANEVTOVERRUN	8	オーバーランエラーが発生し受信データを取りこぼした	
パラメータ	①	<デバイス番号>	数値
	「CANOPEN」でオープンしたデバイス番号を指定します。		
備 考	本コマンドを使用する際は、INCLUDEコマンドでIML002CAN.AJNをインクルードしてください。		
使用例	「2.6 割り込み機能」のプログラム例を参照してください。 ' 識別子の定義の為に必要 INCLUDE "IML002CAN.AJN" ' 割り込み状態を取得します STATUS = CANEVENT(1) ' 受信割り込みだったら、受信処理を行うように分岐します IF STATUS = CANEVTRECV THEN PRINT CANRECV(1) END IF		

3.ON CAN CALL

命令			
機 能	割り込み発生時に、呼び出されるサブルーチンを定義します。 本命令の呼び出し後、「CAN ON」を呼び出すと割り込み有効になります。		
書 式	ON CAN <①デバイス番号> CALL <②サブルーチン名>		
パラ メータ	①	<デバイス番号>	数値
	「CANOPEN」でオープンしたデバイス番号を指定します。		
	②	<サブルーチン名>	サブルーチン名
	要因発生時に実行するサブルーチン名を指定します。 サブルーチンの定義は、以下のようになしてください。 <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> SUB サブルーチン名(<デバイス番号>, <割り込み要因>) 処理内容 END SUB </div> 割り込みが発生すると、検知したCANのデバイス番号と割り込み要因が積み込まれて呼び出されます。		
備 考	<ul style="list-style-type: none"> ・ 割り込みの設定は無効がデフォルトになります。 ・ 割り込みが無効状態時のみ実行できます。 ・ 本命令を呼び出し後、「CAN ON」で割り込みが有効になります。 		
使用例	「2.6 割り込み機能」のプログラム例を参照してください。		

4.CAN ON

命令			
機 能	割り込みを有効にします。		
書 式	CAN ON <①デバイス番号>		
パラ メータ	①	<デバイス番号>	数値
	「CANOPEN」でオープンしたデバイス番号を指定します。		
備 考	<ul style="list-style-type: none"> ・割り込みを有効にし、保留にしていた割り込みを発生させます。 ・本コマンドの実行後、「CANEVENTCONFIG」で有効化した割り込み要因が発生すると、「ON CAN CALL」の実行時に指定されたサブルーチンが実行されます。 		
使用例	「2.6 割り込み機能」のプログラム例を参照してください。		

5.CAN OFF

命令			
機 能	割り込みを無効にします。 これ以降、CAN系の割り込みが検知されても、サブルーチンへの分岐は起こりません。		
書 式	CAN OFF <①デバイス番号>		
パラ メータ	①	<デバイス番号>	数値
	「CANOPEN」でオープンしたデバイス番号を指定します。		
使用例	「2.6 割り込み機能」のプログラム例を参照してください。		

6.CAN STOP

命令			
機 能	割り込みを保留にします。 これ以降、CAN系の割り込みが保留されます。 サブルーチンへの分岐は起こりません。 「CAN ON」で割り込みが許可されると、保留されていた割り込みでサブルーチンに分岐します。		
書 式	CAN STOP <①デバイス番号>		
パラ メータ	①	<デバイス番号>	数値
	「CANOPEN」でオープンしたデバイス番号を指定します。		
備 考	本コマンドの実行後に「CAN ON」を実行すると、割り込みが有効化されるとともに、保留されていた割り込みイベントが実行されます。		
使用例	「2.6 割り込み機能」のプログラム例を参照してください。		

3.7 エラーコード一覧

エラーコードが&h01000012(=ドライバまたはライブラリの呼び出しに失敗した) の時、ERRSUB関数で ドライバ内部のエラーコード値を取得できます。エラーコードの詳細は、以下の表のようになります。

「ERRSUB」で値を受け取るには、「ON ERROR CALL」で、エラー発生時の割り込みルーチン内で、以下のよう記述可能です。

```
' エラーが発生したら、CB_ERR サブルーチンを呼び出すように定義します
ON ERROR CALL CB_ERR
ERROR ON

SUB CB_ERR(I_ERR, I_ERM$, I_ERL)
' ドライバまたはライブラリの呼び出しに失敗した
IF I_ERR = &H01000012 THEN
    NUM& = ERRSUB ' ドライバ内部のエラーコード値を取得します
    PRINT "CAN 内部のエラーコード値="; HEX$(NUM&)
END IF
END SUB
```

値	意味	対処方法
2	CANドライバが見つからなかったかオープンできませんでした。	CANドライバ(GPG-4851)をインストールしているか確認してください。
22	デバイス名の不正などにより、ドライバをオープンできませんでした。	デバイス番号のデバイスが正しく認識されている事を確認してください。
&hC0000001	現在ドライバを使用することができません。	デバイスがちゃんと接続されていることを確認してください。
&hC0000003	パラメータが不正です。	引数の値が不正です。 引数の値の範囲が指定範囲外です。
&hC0000006	メモリを確保できません。	ドライバ内部のバッファメモリの確保に失敗しました。
&hC0000007	バッファに空きがありません。	送信バッファの空き容量以上のデータを送信しようとしてしました。 送信バッファサイズと、送信バッファ内にあるデータ数を確認して、関数を実行してください。
&hC0000008	バッファが空です。	受信バッファ、エラーバッファが空でないことを確認して、関数を実行してください。
&hC0000009	バスが有効時にしか実行できない関数をバスオフ時に実行しました。	バスの状態を確認後、実行してください。
	バスオフ時にしか実行できない関数をバスが有効なときに実行しました。	
	周期送信データのバッファ転送中です。	
&hC000000A	デバイスハンドルが正しくありません。	不正なデバイスハンドルで呼び出しを行おうとしてしました。
&hC000000B	サポートしていない機能です。	サポートしていない機能と呼び出そうとしてしました。
&hC000000C	現在、周期送信中です。	周期送信を停止させてから関数を実行してください。
	現在、データ送信中です。	データが送信完了してから関数を実行してください。

第4章 サンプルプログラム

サンプルプログラムは「/usr/share/interface/AJANPro/samples/IML/CAN/」に格納されています。

AJAN統合開発環境を起動すると、左ペインのエクスプローラウィンドウ内の「Samples/IML/CAN/」に、ファイルが取り込まれて配置されます。

4.1 サンプルプログラム一覧

No.	ファイル名	内 容
1	CANSEND.AJN	CAN通信の送信を行うサンプルプログラムです。 CANRECV.AJNを実行開始から、別の端末でCANSEND.AJNを実行します。 CANOPEN命令で、デバイス番号1に対してオープンした後、CANACTIVATE命令で、CANバスへの接続を有効化し、CANSEND命令で、1件のメッセージを送信します。 その後、10秒程度待機した後、CANDEACTIVATE命令で、接続を解除し、CANCLOSE命令でクローズして終了します。
2	CANRECV.AJN	CAN通信の受信を行うサンプルプログラムです。 CANRECV.AJNを実行開始から、別の端末でCANSEND.AJNを実行します。 CANOPEN命令で、デバイス番号1に対してオープンした後、CANCLEAR命令で、ドライバ内の送受信バッファ等をクリアし、CANACTIVATE命令で、CANバスへの接続を有効化します。 次に、ON CAN CALL命令とCANEVENTCONFIG命令で、データ受信したら割り込みルーチン(CANCALLBACK)を呼び出すようセットし、DO WHILE - LOOPで、待ち続けます。 ここで、別の端末から (CANSEND.AJNを実行するなど) CANメッセージを送信する事を期待しています。 CANデータを受信すると、CANCALLBACK ルーチンが呼び出され、受信割り込みを検知したら、フラグ(LoopFlag)を立てて、DO WHILE - LOOP を抜けるようにします。 ループを抜けた後、CANRECV関数で 1件のメッセージを受信し、受信した内容を表示した後、CANDEACTIVATE命令で、接続を解除し、CANCLOSE命令でクローズして、終了します。
3	CANFDSEND.AJN	CANSEND.AJNの送信処理のCAN FD通信版です。 CANFDRECV.AJNを実行開始から、別の端末でCANFDSEND.AJNを実行します。 1件のメッセージ送信を、CANSEND命令でなく CANFDSEND命令で送信しています。 それ以外は、CANSEND.AJNと同じ動作です。
4	CANFDRECV.AJN	CANRECV.AJNの受信処理のCAN FD通信版です。 CANFDRECV.AJNを実行開始から、別の端末でCANFDSEND.AJNを実行します。 1件のメッセージ受信を、CANRECV関数でなく CANFDRECV関数で受信しています。 それ以外は、CANRECV.AJNと同じ動作です。
5	CANSEND_RECV.AJN	CAN通信の送信・受信のどちらを行うか選択できるサンプルプログラムです。 実行開始すると、1=CAN送信 / 2=CAN受信 / 99=終了 を選択入力します。

		<p>CAN送信を選ぶと、凡そ 1ms周期で乱数値をCAN送信します。</p> <p>CAN受信を選ぶと、受信バッファを確認する都度、受信データがあればCAN受信して、前回の受信との周期性を表示します。</p> <p>CAN受信は、割り込みを用いず、CANSTATUS関数で受信バッファに受信データが入ってきたか否かで、受信可能を判断しています。</p>
6	CANFDSENDRECV.AJN	<p>CAN FD通信の送信・受信のどちらかを行うか選択できるサンプルプログラムです。</p> <p>実行開始すると、最初にデバイス番号を入力した後、1=送信→受信 / 2=受信→送信 を選択入力します。</p> <p>1=送信→受信を選ぶと、CANFDSEND命令でメッセージを送信した後、CANFDRECV関数でメッセージを受信し、これを5回繰り返します。</p> <p>2=受信→送信を選ぶと、CANFDRECV関数でメッセージを受信し、CANFDSEND命令でメッセージを送信し、これを5回繰り返します。</p>

第5章 索引

C	
CAN OFF.....	28
CAN ON	28
CAN STOP	28
CANACTIVATE	15
CANCLEAR.....	16
CANCLOSE.....	15
CANDEACTIVATE.....	15
CANEVENT.....	26
CANEVENTCONFIG.....	25
CANFDRECV	20
CANFDSEND.....	18
CANOPEN	13
CANRECV	19
CANRECVERROR.....	21
CANSEND	17
CANSTATUS	24
O	
ON CAN CALL.....	27

第6章 重要な情報

保証の内容と制限

弊社は本ドキュメントに含まれるソースプログラムの実行が中断しないこと、またはその実行に誤りが無いことを保証していません。

本製品の品質や使用に起因する、性能に起因するいかなるリスクも使用者が負うものとします。

弊社はドキュメント内の情報の正確さに万全を期しています。万一、誤記または誤植などがあった場合、弊社は予告無く改訂する場合があります。ドキュメントまたはドキュメント内の情報に起因するいかなる損害に対しても弊社は責任を負いません。

ドキュメント内の図や表は説明のためであり、ユーザ個別の応用事例により変化する場合があります。

著作権、知的所有権

弊社は本製品に含まれるおよび本製品に対する権利や知的所有権を保持しています。

本製品はコンピュータ ソフトウェア、映像/音声(例えば図、文章、写真など)を含んでいます。

医療機器/器具への適用における注意

弊社の製品は人命に関わるような状況下で使用される機器に用いられる事を目的として設計、製造された物では有りません。

弊社の製品は人体の検査などに使用するに適する信頼性を確保する事を意図された部品や検査機器と共に設計された物では有りません。

医療機器、治療器具などの本製品の適用により、製品の故障、ユーザ、設計者の過失などにより、損傷/損害を引き起こす場合が有ります。

複製の禁止

弊社の許可なく、本ドキュメントの全て、または一部に関わらず、複製、改変などを行うことはできません。

責任の制限

弊社は、弊社または再販売者の予見の有無にかかわらず発生したいかなる特別損害、偶発的損害、間接的な損害、重大な損害について、責任を負いません。

本製品(ハードウェア、ソフトウェア)のシステム組み込み、使用、ならびに本製品から得られる結果に関する一切のリスクについては、本製品の使用者に帰属するものとします。

本製品に含まれる不都合、あるいは本製品の供給(納期遅延)、性能もしくは使用に起因する付随的損害もしくは間接的損害に対して、弊社に全面的に責がある場合でも、弊社はその製品に対する改良(有償サービスの利用)、代品交換までとし、製品の予防交換並びに、代金減額等、金銭面での賠償の責任は負わないものとします。

本製品は、日本国内仕様です。

商標/登録商標

本書に掲載されている会社名、製品名は、それぞれ各社の商標または登録商標です。

改訂履歴

Ver.	年 月	改 訂 内 容
0.90	2019年10月	新規作成
1.00	2022年1月	最新情報に更新

このマニュアルは、製品の改良その他により将来予告なく改訂しますので、予めご了承ください。