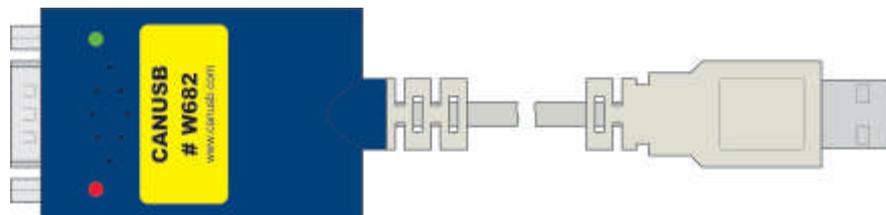




CANUSB

マニュアル



CANUSB manual

2011年11月
Version 1.0D

このマニュアルではそのように明確に示されていない、著作権で保護された製品に対する記載があります。著作権©記号の欠如は、製品が保護されていないことを暗示してはなりません。更に、登録された特許と商標をこのマニュアルでは同様にはっきりと示しています。

この文書の情報は非常に慎重にチェックされ、信頼できると信じています。しかし、LAWICEL 社はいかなる不正確さに対しても責任を負いません。LAWICEL 社は、このマニュアルあるいはその関連製品の使用から生じた損害に対し、保証も与えず、何らの責任も認めません。LAWICEL 社は、前もっての通知なしに、ここに含まれる情報を変更する権利を保有します。

更に、LAWICEL 社は、ここに記載したハードウェアの不適切な使用、または不適切に組み付けたために起こる損害に対して保証を提供するものではなく、いかなる義務も負いません。最後に、LAWICEL 社は前もっての通知なしにハードウェアを変更したり、設計する権利を保有し、そのようにすることに対する責任を認めません。

Copyright 2005–2006 **LAWICEL AB**

All rights reserved. Printed in Sweden.

翻訳、再版、放送、写真製版または同様の複製を含みます。
LAWICEL社からの同意書なしに再生を行うことはできません。

LAWICEL AB
Klubbgatan 3
S-282 32 Tyringe
SWEDEN
Phone: +46 451 59877
FAX: +46 451 59878
<http://www.canusb.com/>
info@canusb.com

1.0 はじめに

LAWICEL CANUSBは低コストで、使い易いCAN/USB dongleです、2種類のドライバーをWindowsまたはLinux O/Sで使うことができます。一つは仮想COMポートドライバ (VCP) で標準RS232 COMポートとして動作します。(このモードでは、CANUSBが当社のCAN232に置き換えられ、わずかな、またはまったくソフトウェアの変更無しにCANのより広い帯域幅を得ることができます)。他のドライバーは、CANUSBと通信するためにDLLを使用するDirect Driver (D2XX) です。両方のドライバーは同じPC上に共存することはできず、それらのうちの一つを選ばなくてはなりません。しかしアンインストールして、例えばVCPからD2XXドライバーに変更するのは簡単です。CANUSBはCAN232 V2のような同じタイプのコマンド (標準ASCII形式) を用いますが、通信の一部の旧方式 (V1の PollやPoll All) は新しい最適化された転送速度のため、取り除かれました。CANUSBは、29ビットID形式と同じく11ビットID形式の両方を取り扱い、RTRフレーム、ビルトインFIFOキュー、拡張info/error情報、わずかのコマンドによる簡単なパワーアップを取り扱い、外部電源は必要でなくUSBポートから供給されます。CANUSBは、ボードの両側に小さなSMD部品の最新技術を使用して、長さ60mm、幅35mm、厚さ15mmしかなく、Atmel AVR, Philips SJA1000 CAN コントローラと FTDI Parallel FT245BM USB チップを陰の力としており、それは高速バスのCANフレームの小さいバーストを扱う方法において非常にフレキシブルにします。この文書は、USBによるブートローダーで更新できるCANUSBの第1バージョンを記載しています。

1.1 CANUSB仕様

- ・長さ:55mm, 幅:36mm, 高さ:16mm
- ・青色透過ケース、1mシールドUSBケーブル
- ・工業用温度範囲 -40°C~+85°C
- ・USB2.0フルスピード, FTDI FT245RL仕様
- ・フィリップス社 SJA1000 CANコントローラ 16MHZ
- ・フィリップス社 82C251 CANトランシーバー (ISO 11898-24V完全互換)
- ・CAN2.0A (11bit ID), CAN 2.0B (29bit ID) 準拠, RTRフレーム対応
- ・無償のDLL, ActiveXインターフェイス, 又はUSBドライバー経由の簡単なアスキーコマンド
- ・CiA DS102-1規格に基づくCANバス接続
- ・USBからの給電 (5V), 外部電源必要なし
- ・CE承認済, ROHS, WEEE準拠

1.2 インストール

最新のドライバーパッケージ (VCPとD2XX) については、www.canusb.comを参照してください。CANUSBサイトに、ドライバーに関してCANUSBの詳しいインストール説明があります。最もニーズに適したドライバーの一つを選択して下さい。CAN232をCANUSBに置き換えることが最もご希望に合っているかもしれません。その場合、VCPドライバーを選べば素早く実行できます。D2XXドライバーは、CANフレームの移動では速いけれども、全体として異なるアプローチを必要とします。CANUSBはUSBコネクタ (USB標準5V) から給電されますので、どんな外部電源も必要ありません。

1.3 CANUSBの試験

この試験はVCPドライバーを使っていると仮定します。必要なら、前のページの1.2節の導入に従ってドライバーをインストールし、CANUSBをPCのUSBポートにインストールすることによって試験して下さい。CANUSBがパワーを得ると、2つのLED(赤と緑)が、それをインストール後(それを接続後約1秒後)に3度素早く点滅します。そして、Windows Terminalソフトウェア(あるいはあなたの好きなターミナルソフトウェア)を起動し、何らかのボーレート、8データビット、パリティなし、1ストップビットに設定し、また、あなたが入力するものをあなたが見られるようにローカルエコーをセットし、それを受信したときと、ラインの終わりに、改行を追加するようにチェックフラグをセットします。最後に、あなたがハードウェアとソフトウェアハンドシェイクをオフにしたことを確認します。そして、接続していることを確認し、>ENTER< を押してください、新しい行を作成するでしょう、それから、Vと>ENTER<を押してください、それがVhhssを印刷します、ここではhhがハードウェアのバージョンで、ssがソフトウェアのバージョンです。あなたは、CANUSBユニットで完全な接続をし、CAN速度を設定し、CANポートをオープンし、フレームを送受信できます。緑のLEDが、CANフレームがうまく送られるか、CANUSBユニットに受信されたことを示します。CANフレームを送受信するために、少なくとも2ノード(CANUSBは1ノードとして数えられる)を持たねばならないことと、CANケーブルネットワークがCANLとCANHライン上の両端を120オームで終端されること、その上、ツイストペアCANケーブルを用いていることにご注意下さい。CANUSBは全フレームを受け入れるようにセットされているので、試験のためにフィルターなどのセットは必要ありません。CANUSBは、www.canusb.comのサンプルプログラムでも試験できます。

1.4 CANUSBの限界

もちろん、CANUSBが如何に多くのフレームを送受信できるかの限界があります。現在のバージョン(V1011)を、125kビットCANビットレートの8データバイトとVCPドライバーで1000スタンダード11ビットフレーム送信のスループットにより試験します。もちろん、ボトルネックは、秒当たり多くのフレームを処理できないRS232 VCPドライバーとマイクロコントローラです。そして、CANUSBは低速のCANに狙いを定められて、250kビットかそれ以下のCAN速度で非常にうまく作動しますが、それは1Mビット(しかし、バスロードをこの速度では高くできない、あるいは、例えばフィルターをフレームの一部を受け入れるようにセットしなければならない)まで使用できます。CANUSBは送信と受信のための、ソフトウェアCAN FIFOを持っています。これらの送信FIFOは8フレーム(スタンダードあるいは拡張)を取り扱うことができ、一方、受信FIFOは32フレーム(スタンダードあるいは拡張)を取り扱うことができます。更に、CANUSBはハードウェアにビルトインされたUSB FIFOも持つので、1度に1つか2つのコマンドだけを扱い、それは次のコマンドの送信前に、CANUSBユニットからの答え(OKである[CR]あるいはエラーの[BELL])を待たなければならないことを意味しております。

1.5 CANドライバー設計ガイド

CANUSBは高度なCANドライバー/構文解析ツールと共に納入されません。多くの市販開発ツールがRS232 ASYNC LIB(Visual Basic、Delphiのような)を提供しているため、VCPドライバーでCANUSBユニットと”talk”するための簡単なプログラムを書くことは容易です。最良の方法はCANUSBへの全ての通信を扱い、アプリケーションに従って全てのメッセージをFIFOまたはメールボックスに置くスレッドを作ることです。転送が速いですが、D2XXドライバーの使用は類似した方策を持っており、COMポートへの”talking”の代わりに、C、C++、VBあるいはDelphiなどでその代わりに使用できるDLLへのAPI (Application Programmer Interface) を持っています。

あなたがCAN232から移っている場合は、CANUSBは古い形のPとAコマンドを持っていません。その代わりに、入ってくるCANフレームをすぐにUSBへ送り出します。これはドライバーのUSB側の帯域幅の節約です。t、T、rとRコマンドからの返答は常に、コマンドに依存してz[CR] あるいは Z[CR]を答えます。その他のコマンドは同じです。

常に、全ての前のコマンドあるいはCANUSBの待ち行列に入った文字(電源投入時、何度も、待ち行列に間違った文字か、前のセッションからの古いもの)を空にするために、2-3の[CR]を送信して、各セッション(プログラム起動時)を起動します、VコマンドでCANバージョンをチェックし(正しい速度でユニットと通信していることを確認するため)、sあるいはSコマンドでCAN速度を設定し、0でCANポートをオープンして、CANUSBがCANフレームの送受信の両方に対し作動中となります。tあるいはTコマンドによりフレームを送信し、それがCAN FIFOに置かれるか、あるいは列が一杯かを見るために返答を待ちます。CANバスから入ってくるフレームはすぐに送り出します。そして、時々(例えば、500-1000ms毎あるいはCANUSBからエラーを戻された) エラーがあるか見るためにFコマンドを送ります。もし、ユニットにコマンドを送った後に多くのエラー応答を得た場合、バッファを空にするために2-3の[CR]を送信します、コマンドを再び発行し、これが続く場合、通信エラーがある(例えば、CANトランスミッターの損傷や電源障害など)ことを、プログラム内でユーザーあるいはアプリケーションに警告します。www.canusb.comウェブサイトはソースコードで多くのサンプルプログラムを提供します。これらのプログラムは使用が無料か、あなたのニーズに適するように変わります。

1.6 バージョン情報

CANUSBのバージョンナンバーは、ハードウェアに対するものとソフトウェアに対するものの2つのバージョンから構成されます。これらの2つのバージョンナンバーは、Vで始まり、ハードウェアに対する2文字とソフトウェアに対する2文字の5文字で1つの固有のバージョン文字列に組合せられています。例えば、バージョンV1010は、それがハードウェアバージョン1.0で、ソフトウェアバージョン1.0であることを示します。もし、ハードウェアを更新すると、2つの最初の文字のバージョン数を増加し、コマンドの追加変更やバグ訂正をする場合は、ソフトウェアバージョンナンバーを増加します。CANUSBがこのマニュアルのコマンドをサポートしているかどうかを見るためには、CANUSBにVコマンドを送ることによって、どのバージョンを持っているかをチェックします(下記のコマンドでそれが如何に作動するかを見ます)。

2.0 使用可能なCANUSB ASCIIコマンド:

注: CANUSBへの全てのコマンドは[CR] (Ascii=13) で終わらなければなりません。
大文字と小文字は区別されます。

Sn[CR] 標準CANビットレートの設定、nは0-8。
このコマンドはCANチャンネルがクローズの時のみ作動します。

S0	10Kbitに設定
S1	20Kbitに設定
S2	50Kbitに設定
S3	100Kbitに設定
S4	125Kbitに設定
S5	250Kbitに設定
S6	500Kbitに設定
S7	800Kbitに設定
S8	1Mbitに設定

例: S4[CR]

CANを125Kbitに設定します。

戻り: OKに対しCR (Ascii 13) またはERROR(エラー)に対しBELL (Ascii 7)。

sxxyy[CR] BTR0/BTR1によるCANビットレートの設定、xxとyyは16進数。
このコマンドはCANチャンネルがクローズの時のみ作動します。

xx	16進数のBTR0の値
yy	16進数のBTR1の値

例: s031C[CR]

125Kbitに等しいBTR0=0x03とBTR1=0x1CでCANを設定します。

戻り: OKに対しCR (Ascii 13) またはERROR(エラー)に対しBELL (Ascii 7)。

O[CR] CANチャンネルをオープン。
このコマンドはCANチャンネルがクローズの時に、直前にSまたはsコマンドが
セットされている時のみ作動します。

例: O[CR]

CANチャンネルをオープンします。

戻り: OKに対しCR (Ascii 13) またはERROR(エラー)に対しBELL (Ascii 7)。

C[CR] CANチャンネルクローズ

このコマンドはCANチャンネルがオープンの際のみ作動します。

例 : C[CR]

 CANチャンネルをクローズします。

戻り : OKに対しCR (Ascii 13) またはERROR (エラー) に対しBELL (Ascii 7)。

tiildd...[CR] スタンダード (11bit) CANフレームの送信

このコマンドはCANチャンネルがオープンの際のみ作動します。

iii 16進数 (000-7FF) のID

l データ長 (0-8)

dd 16進数 (00-FF) のバイト値、dd対の数はデータ長に合わなければなりません、さもないとエラーが発生します。

例 1 : t10021133[CR]

 値0x11と0x33の2バイトで、ID=0x100の11ビットCANフレームを送信。

例 2 : t0200[CR]

 ID=0x20と0バイトで11ビットCANフレームを送信。

戻り : OKに対しz[CR] またはエラーに対しBELL (Ascii 7)。

Tiiiiiiidd...[CR] 拡張 (29bit) CANフレームの送信

このコマンドはCANチャンネルがオープンの際のみ作動します。

iiiiiii 16進数 (00000000-1FFFFFFF) のID

l データ長 (0-8)

dd 16進法 (00-FF) のバイト値、ddの数はデータ長に合わなくてはなりません、さもないとエラーが発生します。

例 1 : T0000010021133[CR]

 値0x11と0x33の2バイトで、ID=0x100の29ビットCANフレームを送信。

戻り : OKに対しZ[CR] あるいはエラーに対しBELL (Ascii 7)。

riiii[CR] スタンダード(11bit)フレームのRTR を送信
このコマンドはCANチャンネルがノーマルモードでオープンの際のみ作動します。
iii 16 進のID(000-7FF)
l データ長(0-8)
例1 : r1002[CR]
ID=0x100, データ長2byte の11bit RTR フレームの送信
戻り : Auto Poll がオフ(デフォルト)であれば、OK に対しCR(Ascii 13)または
ERROR(エラー)に対しBELL(Ascii 7)。Auto Poll がオン (X コマンド参照) であれば、
OK に対しz[CR]またはERROR に対しBELL(Ascii 7)。

Riiiiiiii[CR] 拡張(29bit)フレームのRTR を送信
このコマンドはCANチャンネルがノーマルモードでオープンの際のみ作動します。
iiiiiii 16 進のID(00000000-1FFFFFFF)
l データ長(0-8)
例1 : R000001002[CR]
ID=0x100, データ長2byte の29bit RTR フレームの送信
戻り : Auto Poll がオフ(デフォルト)であれば、OK に対しCR(Ascii 13)または
ERROR(エラー)に対しBELL(Ascii 7)。Auto Poll がオン (X コマンド参照)
であれば、OK に対しZ[CR]またはERROR に対しBELL(Ascii 7)。

F[CR] ステータスフラグ読み出し
このコマンドはCANチャンネルがオープンの際のみ作動します。
例 1 : F[CR]
 ステータスフラグを読みます。
戻り : OKに対し、Fと2バイトBCD16進数値プラスCR (Ascii 13)。もしCANチャンネル
 がオープンしていないと、BELL (Ascii 7)を返す。このコマンドもRED Error
 LEDをクリアします。下の利用可能なエラーをご参照下さい。
例、 F01[CR]
Bit 0 受信時、FIFOキューがフル
Bit 1 送信時、FIFOキューがフル
Bit 2 エラー警告(EI)、SJA1000データシートをご覧ください
Bit 3 データオーバーラン(DOI)、SJA1000データシートをご覧ください
Bit 4 使われていません
Bit 5 エラーパッシブ(EPI)、SJA1000データシートをご覧ください
Bit 6 アービトレーションロスト(ALI)、SJA1000データシートをご覧ください*
Bit 7 バスエラー(BEI)、SJA1000データシートをご覧ください**
*アービトレーションロストは赤LED点滅を発生しません!
**バスエラーは赤LEDを連続点灯します!

Mxxxxxxx[CR] Acceptance Code Register(SJA1000のACn Register)をセット
このコマンドは、CANチャンネルが初期化され、まだオープンしていないときのみ作動します。

xxxxxxx 16進のAcceptance Code、LSB first、AC0、AC1、AC2、AC3
詳細はPhilips SJA1000データシートをご覧ください。

例: M00000000[CR]
Acceptance Codeを0x00000000にセットします。
これは電源オン時の初期設定です、すなわち、全てのフレームを受信。

戻り: OKに対しCR (Ascii 13)あるいはエラーに対しBELL (Ascii 7)。

mxxxxxxx[CR] Acceptance Mask Register(SJA1000のAMn Register)をセット
このコマンドは、CANチャンネルが初期化され、まだオープンしていないときのみ作動します。

xxxxxxx 16進のAcceptance Mask、LSB first、AM0、AM1、AM2、AM3
詳細はPhilips SJA1000をご覧ください。

例: mFFFFFFFF[CR]
Acceptance Maskを0xFFFFFFFFにセットします。
これは電源オン時の初期設定です、すなわち、全てのフレームを受信。

戻り: OKに対しCR (Ascii 13)あるいはエラーに対しBELL (Ascii 7)。

Acceptance CodeとMaskレジスタの設定

Acceptance Code RegisterとAcceptance Mask Registerと一緒に作動し、それらは2グループのメッセージをフィルタリングできます。これがどのように作動するかの詳細についてはSJA1000データシートをご覧ください。11ビットIDのものでは、この方法で単一IDをフィルタリングできますが、29ビットIDのものでは、IDのグループをフィルタリングできるだけです。下の例は0x300から0x3FFの全ての11ビットIDのものをだけを受信するようにフィルターをセットします。

コマンド	コメント
M00006000[CR]	AC0=0x00, AC1=0x00, AC2=0x60, AC3=0x00
m00001FF0[CR]	AM0=0x00, AM1=0x00, AM2=0x1F, AM3=0xF0

最初のコマンドは、セットされてない場合フィルター2に対して、2ビットを合わせるように告げ、(この場合、それは0x3nn、3に相当)。第2のコマンドはnnに気にならないように告げるので、それらからFFまでにできます、メモリー列に良く配置されていないので、読むのがそれほど簡単ではないが。フィルター1はオフになっています(AM0、AM1とAM3の下半分)。マスクの最後のバイトも0xF0の代わりに0xE0にすることができ、RTRビットをフィルタアウトし、RTRフレームを受け入れます。

V[CR] バージョン番号の取得、CANUSBのハードウェアとソフトウェアの両方
このコマンドは常に作動しています。

例: V[CR]
バージョン番号を得ます。

戻り: OKに対し、Vとハードウェアバージョンの2バイトBCD値と、ソフトウェアの2バイトBCD値プラスCR (Ascii 13)。例えば、V1013[CR]

N[CR] シリアル番号の取得
このコマンドは常に作動します。

例: N[CR]
シリアル番号を得ます。

戻り: OKに対しNとシリアル番号の4バイト値プラスCR (Ascii 13)。
例えば、NA123[CR]
シリアル番号はそこに数値とアルファベットの両方を持っていることにご注意下さい。簡単な参照のために、シリアル番号をCANUSBにも印刷していますが、例えば、CANUSBをプログラムで認識するために使用することができますので、プログラムはそれが正しい方法で設定されていることを知ります (パラメータはEEPROMに保存されています)。

Zn[CR] 受信したフレームのTime Stamp ON/OFFセット

このコマンドは、CANチャンネルがクローズしているときだけ作動します。

この値はEEPROMに保存され、次にCANUSBに電源を入れるまで記憶されます。このコマンドは、次に挙動を変えたい時以外に使用すべきではありません。CANUSB用に書かれた古いプログラムとの互換性のために、初期設定ではOFFにセットされています。ONにセットすることは、AとPコマンドか、Auto Poll/Sendがイネーブルの時に、CANUSBからの送出しに4バイトを追加します。Time Stampを使用すると、各メッセージはそれがCANUSBに受信された時に、ミリ秒の時間を得ます、これは例えば、メッセージ間の時間を知るためのリアルタイムアプリケーションに使用できます。しかし、送られている各メッセージに4バイトを追加するので、この特性の使用によって、多分CANUSBの帯域幅を減らすことに注意して下さい。(特に、VCPドライバーで)

Time StampがOFFならば、入ってくるフレームはこのように見えます:

t10021133[CR] (ID=0x100と2バイトを持つスタンダードフレーム)

Time StampがONであれば、入ってくるフレームはこのように見えます:

t100211334D67[CR] (ID=0x100と2バイトを持つスタンダードフレーム)

最後の4バイト0x4D67は、ミリ秒(もちろん16進数)でのこの仕様のメッセージに関するTime Stampであるにご注意下さい。CANUSBのタイマーはゼロ0x0000で始まり、ループする前に0xEA5Fまで行き、0x0000へ戻ります。これは正確な60,000mS(すなわち、大部分のシステムで十分以上である1分)に一致します。

例1: Z0[CR]
Time Stamp特性をOFFにします。(初期設定)

例2: Z1[CR]
Time Stamp特性をONにします。

戻り: OKに対しCR (Ascii 13)あるいはエラーに対しBELL (Ascii 7)