

CarLink CANusbAccessory : 説明書



[初版] 2012/3/29

[修正] 2013/11/5

1. はじめに

本ドキュメントでは、自動車に搭載のOBD-IIコネクタを介して車両情報を含むCANメッセージをBluetoothにて転送する装置（以下、CANusbAccessory装置）と、Android用ドライブレコーダソフトウェア（以下、CarLink CANusbAccessoryソフトウェア）について説明します。

両装置とソフトウェアを利用することにより、自動車運転時の車両情報、各センサ値、映像と音声といった記録データをAndroid端末上に保存することが可能になります。

1.1 CANusbAccessory装置について

CANusbAccessory 装置は、CAN-BUSから受信したメッセージを、Android USB AccessoryデバイスとしてAndroid端末へ転送します。車両のOBD-IIコネクタ内のCAN-L/CAN-Hラインより受信したCANメッセージを、可能な限りの速度にて転送し、1.0msec間隔にて送信されるメッセージにも対応可能なことが特徴です。

車両への接続は、本装置をOBD-IIコネクタに差し込むだけで完了します。駆動電源は、車両のOBD-IIコネクタを介して、バッテリー（+12V）から確保します。また、接続しているAndroid端末にもMicroUSBポート経由にて電源を供給可能です。

1.2 CarLink CANusbAccessoryソフトウェアについて

CarLink CANusbAccessoryソフトウェアは、Android端末上にて動作し、CANusbAccessory装置の制御と、受信したCANメッセージをSDカードへ保存する機能を持ちます。また、あわせて、Android端末に搭載のカメラとマイクからの映像と音声の記録、加速度センサによる加速度、地磁気センサ（もしくはジャイロセンサ）との組み合わせによる傾きの値、緯度経度情報もSDカードへ保存可能です。

以下では、両ソフトウェアと装置を用いて車両のCANメッセージと映像、音声、各センサ値を記録する方法について、説明します。

2. 利用手順

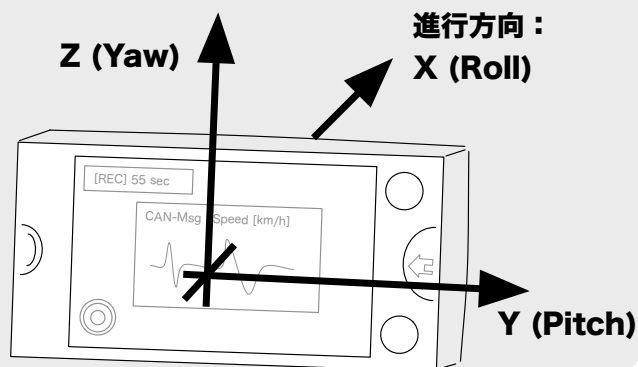
2.1 セットアップと操作方法

(Step 1) Android端末のセットアップ

CarLink CANusbAccessoryソフトウェアを、Android 2.3.4以上の端末へインストールします。

また、OSのバージョンだけでなく、
Android USB Accessoryのサポートが必須です。

CarLink CANusbAccessoryソフトウェアでは、
記録する加速度センサと傾きを、右図の軸にて記録します。
適切に、車両内にAndroid端末本体を固定してください。



(Step 2) CANusbAccessory装置の取り付け、Android端末をCANusbAccessory装置に接続

対象の車両のエンジンが停止状態であることを確認後、車両のOBD-IIコネクタ差し込み口へ、CANusbAccessory装置のOBD-IIコネクタを差し込みます。運転に支障がないように、本装置を固定してください。

Android端末を、CANusbAccessory装置のUSBポートと接続します。

※ CANusbAccessory装置のUSBポートには、Android端末以外を接続しないでください。
故障の原因になります。

(Step 3) Android端末の設定

初回接続時に表示される、CANusbAccessory装置を接続した際のAndroid端末の動作を決定するダイアログにて、常にCarLink CANusbAccessoryアプリを使用するように、チェックボタンを有効化します。

(Step 4) CarLink CANusbAccessoryソフトウェアの起動と記録操作

Android端末にて、SDカードが利用可能状態であることを確認後、CarLink CANusbAccessoryソフトウェアを起動します。

起動後、メニューもしくは、画面左下のボタンより、記録を開始できます。車種別設定ファイル（次節にて詳細を説明）を選択すると、SDカードへ各種データが書き込まれ始めます。再度、同様のボタン操作を行うと記録を停止します。Android端末の戻るボタンもしくは、ホームボタンを押した場合も、記録を停止します。

車種別設定ファイルは、画面上に、受信したCANメッセージのバイト列を換算して得られる車両情報を、グラフとして表示するためのものです。車種別設定ファイルを未設定の場合は、「利用なし」を選択することも可能です。車種別設定ファイルを利用しない場合、画面上にCANメッセージのトラフィックしか表示されませんが、車種別設定の利用有りの場合と同様に、SDカードへは、全受信メッセージが保存されます。

記録中は、指で画面を水平方向にスライドすることにより、表示する項目を変更できます。CANメッセージを選択中は、さらに、上下方向へのスライド操作により、車種別設定ファイルに対応する車両情報の項目へ変更できます。

2.2 車種別設定ファイルの記述方法

車種別設定ファイルを記述することにより、CANメッセージの記録中に、画面上に車両情報の遷移グラフを表示することができます。CANメッセージより取得可能な速度や燃費、ステアリング角などといった車両情報に対応するCANメッセージは、車種によって異なるCAN-IDを持つため、記録対象の車両に応じて設定ファイルを記述する必要があります。

車種別設定ファイルの記述形式を以下に示します。

各設定ファイルは、csv形式にて、Android端末をPCと接続した際にマウントされるストレージ（Android 2.x系であればSDカード、Android 4.x系であれば内蔵フラッシュメモリ）の下に置きます。

例えば、Android 2.x系統の端末では、SDカード直下に/carlink_can_bt/configフォルダを作成し、置きます。Android 4.0では、/mnt/sdcardフォルダがマウントされる場合が多いですので、その下に、carlink_can_bt/configフォルダを作成し、置きます。Android 4.1/4.2では、/storage/sdcard0がマウントされる場合が一般的ですので、その下に、carlink_can_bt/configフォルダを作成し、置きます。

1行目： 任意にコメントなどを記述可能な行です。

2行目： 画面表示の対象にするCAN-IDと、後述しますバイト列の換算式の係数を「,」区切りで並べて記述します。
CAN-ID, description, min, max, k, offset, sign_flag, num_bits, i, mask(hex), s, i, mask(hex), s, ...

... 3行目以降： 2行目と同様に、表示対象のCAN-ID毎の記述を繰り返し。

さらに、詳しくCAN-IDと換算式の係数の項目の一覧を以下に示します。

- ・CAN-ID (hhh, 16進3桁)

表示対象のメッセージの種類に対応する識別子CAN-IDを、16進数3桁にて、指定します。

例えば、「0B0 000000001103」という、CAN-IDを持つメッセージを対象にする場合は、CAN-IDに相当する「0B0」となります。

- ・Description（任意の文字列）

Android端末上にて表示する車両情報の名称を文字列にて指定。

- ・最大値max（実数）

グラフ表示用に、換算後の値が取り得る上限値を指定します。

- ・最小値min（実数）

グラフ表示用に、換算後の値が取り得る下限値を指定します。

- ・符号の有無 sign_flag（符号付き = 1, 符号なし = -1）

- ・ビット長 num_bits（1から63までの整数）

次項に示しますhex_valueを、符号付き整数として扱うのか、符号なしの整数として扱うかのフラグです。符号付き整数とする場合は、sign_flagに-1を指定してください。また、符号付き整数の場合、ビット長をnum_bitsに指定してください。

- ・ value/bits係数k（実数）、offset値（実数）、[インデックス番号i（整数）、mask（16進数）、左シフト量s]、[]の組み合わせを必要数だけ繰り返し、...

CANメッセージ中のバイト列に対する換算は、次式のように行います。

$$\text{hex_value} = ((\text{CANmsg_bytes}[i] \& \text{mask}) \ll s) + \dots$$

$$\text{value} = k * \text{hex_value} + \text{offset}$$

CANメッセージ中のバイト列を、i, mask, s, i, mask, s, ... の指定値にしたがい、並び替え、結合し、hex_valueに変換します。

バイト列CANmsg_bytesのi番目に対して、maskをAND演算し、sの分だけ左シフトする演算（sが負値の場合は、右シフト）を、必要なインデックスのバイトデータに対して行った結果の総和に、1bit当たりの量の係数kを掛け、offsetを加算。 バイトデータのインデックス番号は0から開始ですのでご注意ください。

hex_valueの値は、sign_flagに応じて必要であれば、num_bitsビット長の符号付き整数として扱います。その後、kとoffsetを用いて、実数へ換算します。最終的な換算値は、上式のvalueです。

例えば、「0B0 11 22 33 44 55 66」というCAN-ID = 0x0B0のメッセージに対して、min=0, max=90にて、k=1.0, offset=0.0, 符号なし8ビット長の値にて、0番目と1番目のバイトデータを並べた値を表示したい場合、「0B0, 項目名称[単位],0,90,1.0,0.0,1,8,0,FF,8,1,FF,0」のように書きます。

設定ファイル全体の具体例を以下に示します。

(vitzにおける例)

```
CAN-ID(hhh),description,min,max,k,offset,sign_flag,num_bits,i,mask(hex),s,i,mask(hex),s, ...
0B0,Speed [mph],0,100,1.0,0.0,1,16,0,FF,8,1,FF,0
2D0,Engine RPM [rpm],0,20000,1.0,0.0,1,16,0,FF,8,1,FF,0
610,Short term fuel efficiency,0,100,1.0,0.0,1,8,2,FF,0
2C1,Throttle position,0,256,1.0,0.0,1,16,6,FF,8,7,FF,0
224,Breake,0,1,1.0,0.0,1,8,0,FF,0
```

(crownにおける例)

```
CAN-ID(hhh),description,min,max,k,offset,sign_flag,num_bits,i,mask(hex),s,i,mask(hex),s, ...
498,Speed,0,100,1.0,0.0,1,8,5,FF,0
```

2D0,Engine RPM [rpm],0,20000,1.0,0.0,1,16,0,FF,8,1,FF,0
610,Short term fuel efficiency,0,100,1.0,0.0,1,8,2,FF,0
025,Steering,0,1000,1.0,0.0,1,16,1,FF,8,7,FF,0
2C1,Throttle position,0,256,1.0,0.0,1,16,6,FF,8,7,FF,0
224,Breake(ON/OFF),0,1,0.05,0.0,1,8,0,FF,0
548,Breake,0,10000,1.0,0.0,1,16,4,FF,8,5,FF,0

3. ログファイル

ログファイルの構成と形式について説明します。

CarLink CANusbAccessoryソフトウェアでは、CANメッセージ、および、映像と音声、Android端末の各センサ値を、それぞれ専用のファイルへ書き出します。複数のファイルへ分けて保存する理由は、データのソース毎に取得タイミング、および、周期が異なるためです。

ログファイルの基本構成は、全種類のデータにおいて共通となっています。csv形式にて、CANusbAccessory装置がCANメッセージを受信した時点のタイムスタンプと、対応する値もしくはバイト列を含む行が続きます。タイムスタンプは2種類、記録されます。現在時刻とUTC1970年1月1日 0時からの差をmsecにて示した32ビット整数型に換算した値と、参考情報としてCANusbAccessory装置内部の μsec でのカウンタ値です。

ログファイルの格納場所は、Android 2.x系の端末では、SDカードの/carlink_can_bt/log/（記録開始日時） / フォルダ下です。Android 4.0では、内蔵フラッシュメモリの/mnt/sdcard/carlink_can_bt/log /（記録開始日時） / フォルダ下です。Android 4.1/4.2では、内蔵フラッシュメモリの/storage/sdcard0 /carlink_can_bt/log/（記録開始日時） / フォルダ下です。ファイル名は、データの種類毎の接頭語に開始日時をつけ加えたものです。

以下では、データの種類毎に詳細を説明します。

・CANメッセージ（ファイル名：cl_can_日時.csv）

「タイムスタンプと、CANメッセージ」を記録します。

例：

```
SysTimeStamp[ms],TimeStamp[us],ID(hhh),DataLength(h),Data(hh...)
1333000216492,07495776,0ba,3,0207c60000000000
1333000216497,07501586,0b0,6,00000000110e0000
....
```

メッセージ長が、8バイト未満のデータは、先頭からメッセージ長までのバイト数が有効なデータです。残りは、00で埋められた値です。

・加速度（ファイル名：cl_acc_日時.csv）

Android端末上の加速度センサの値は、「タイムスタンプ、X軸方向の加速度[G]、Y軸方向、Z軸方向」の並びにて、記録します。

サンプリング周期は、Android-OSと端末本体の実装に依存します。参考までに、中程度の優先度を設定した場合にて、SonyEricsson Xperia X10 (Android 2.1)では約30msec間隔 \approx 33Hz です。

例：

```
TimeStamp[ms],X[G],Y[G],Z[G]
1314778499364,-6.9671874,-7.4265623,2.0671875
1314778499394,-6.7375,-7.4648438,1.8757813
1314778499424,-6.584375,-7.3882813,1.990625
...
```

・姿勢（Android端末の傾き）（ファイル名：cl_pose_日時.csv）

姿勢は、加速度センサと地磁気センサを用いて求めた傾き値を記録します。値の並びは、「タイムスタンプ, Roll周りの姿勢[rad], Pitch, Yaw」となります。単位は、[rad]です。

（ジャイロスコプ搭載の端末では、角速度の積算にて姿勢を求めることも可能です。）

例：

```
TimeStamp[ms],Roll[rad],Pitch[rad],Yaw[rad]
1314778499331,-3.1415927,0.0,0.0
1314778499431,0.6198268,0.82110023,1.2772074
1314778499531,0.63624763,0.84663844,1.2471836
...
```

・位置（ファイル名：cl_loc_日時.csv）

位置は、緯度、経度を「タイムスタンプ, 緯度, 経度」の並びにて記録します。サンプリング周期は、1Hz程度になるように調整しています。

・映像と音声（cl_video_日時.mp4）

Android端末に搭載のカメラとマイクより記録したビデオファイルは、mpeg4形式にて保存します。解像度はQVGAです。