

車両 CAN バス インターフェース RLVBCAN02

操作マニュアル

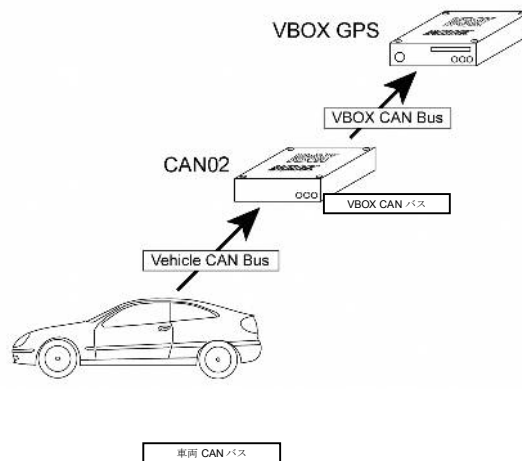


目次

目次	2
はじめに	3
主な特徴	3
RLVBCAN02 付属パーツ	4
別売アクセサリ	4
仕様	4
CAN に関する解説	5
CAN02 と VBOX との接続	7
セットアップ	8
CAN02 のセットアップ	8
シリアルデータ	10
信号パラメータの解説	11
CAN データベースの作成	15
ピン配列	16
コネクタ 1 - CAN / 電力入力	16
コネクタ 2 - CAN / 電力出力	16
コネクタ 3 - ファームウェアアップグレード	17
コネクタ 4 - 車両 CAN バス	17
LEMO から 9 ピン D への RLVBCAB19 ケーブル	18
モジュール寸法	19

はじめに

CAN02(RLVBCAN02) は、Racelogic 社の VBOX データロガーに車両 CAN バスを取り込むことを目的に設計されています。記録する CAN データはユーザーが任意に決定でき、車両から取り込む際のデータのスケールおよびオフセット値を自由に設定することが可能ですので、実質的なデータを記録することができます。



主な特徴

250K ビット、500K ビット、および 1M ビット のボーレート

最大で 16 個の CAN パラメータの取得

値を記録するためのスケールおよびオフセットの調節が可能

車両 CAN バスへの不要な VBOX データ流出をブロック

Racelogic 社が独自に解析をした、車両 CAN のデータベースの利用が可能

RLVBCAN02 付属パーツ

RLVBCAN02 x 1	車両 CAN BUS インターフェース (VCI)
RLCAB006 x 1	VBOX への接続ケーブル
RLCAB019L x 1	5 ピン LEMO - 9 ピン D サブ ターミナルケーブル

別売アクセサリ

RLCAB001	スタンドアローン設定用シリアル変換ケーブル
RLCAB20	OBD-II - 9 ピン D サブ CAN インターフェースケーブル (注：すべての車両 CAN が、OBD コネクタで取得可能ではありません。)
RLCAB005-C	VBOX への接続ケーブル - 1.5m

仕様

入力チャンネル	車両または他の CAN バス からの CAN データチャンネル (16 チャンネル)
入力データタイプ	1~16 ビット。Intel / Motorola フォーマット。符号付き / 符号なし。スタート ビット 選択可能。
出力チャンネル	VBOX への CAN データチャンネル (x 16)。
出力データタイプ	F4 フローティングポイントデータ。スケールおよびオフセットが調整可能。
電源電圧	+9~+18 V DC
CAN タイプ	CAN 2.0A または CAN 2.0B に適合。
ボーレート	250K ビット/s 500K ビット/s 1M ビット/s
車両バス 側の CAN 端子抵抗	無し
設定	VBOX Tools, CAN02 Setup ソフトウェアを使用

CAN に関する解説

CAN とは、コントローラエリアネットワーク（Controller Area Network）の略語です。CAN は多重配線の形態であり、Bosch 社によってデザインされました。これにより、車両の複数の制御システム（通常は車両内部のもの）を 1 対(2 本)の通信線のみでリンクすることが可能となり、制御システム間での情報共有もできるようになりました。このネットワークが開発される以前は、車両内で各信号に対して少なくとも 1 本のワイヤーが必要であったため、スペースが必要でコストも高くなりがちでしたが、CAN バスの多重通信によって、多数の信号をデジタルで送信できるようになり、また、情報の共有化によって必要なセンサの数も減らすことが可能となりました。例えば、エンジンコントローラは冷却水温度をモニターするためのセンサが装備されています。CAN を活用すれば、このセンサは定期的に冷却水温度を送信することで、他の関連システム、例えばコンビネーションメータ（冷却水温度情報を元にして水温計の針を上下させ）に温度情報を伝えることができます。

CAN は、信号をデータパケットとして送信します。このパケットは「フレーム」と呼ばれ、各フレームは、1 つの識別番号（ID）および一群のデータバイト(byte)で設定されています。各 ID に対し最大 8 バイトを割り当てることができ、一般的には、各 ID に複数の信号が割り当てられています。標準的な CAN ID は 11 ビット長で構成されています。11 ビットでは、4096 個の ID の割り当てが可能です。CAN フレームの例を下に示します。

ID	データ バイト							
	1	2	3	4	5	6	7	8

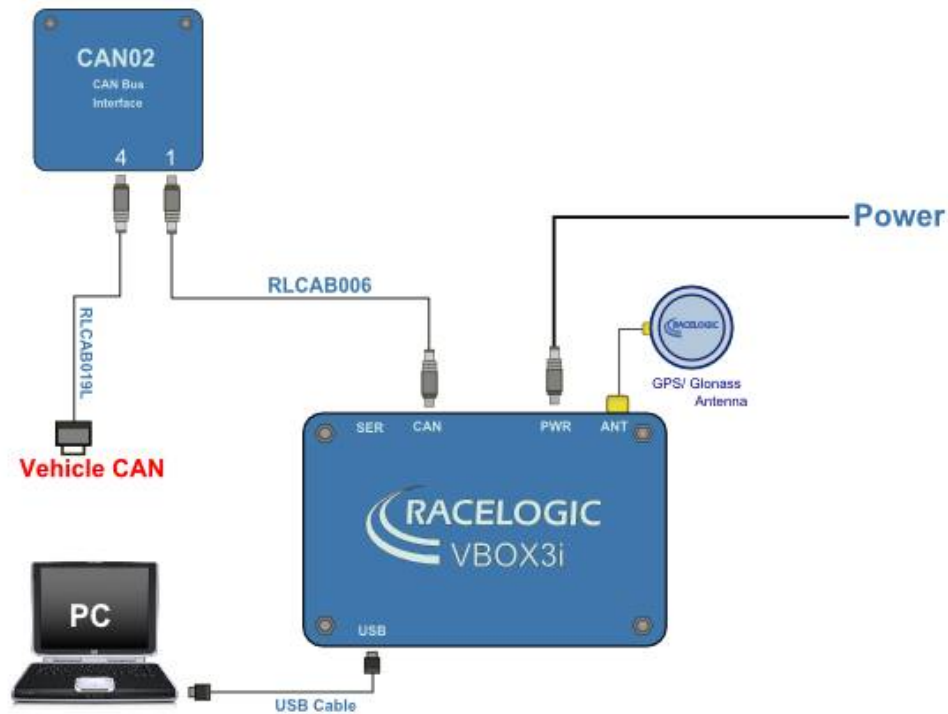
CAN の物理的側面は標準化されていますので各社共通ですが、ID を CAN バス 信号へ割り当てる方法は自動車メーカーにより様々です。つまり、ある自動車メーカーでは RPM（エンジン回転数）の信号に ID “xyz” を割り当てたとしても、別のメーカーでは xyz は使用されていなかったり、もしくは異なった、例えばステアリング角度に割り当てている、ということになります。通常、車両の CAN フレームは次の図のようになります。

ID	データ バイト 数							
	1	2	3	4	5	6	7	8
	↓	↓	↓	↓	↓	↓	↓	↓
432	RPM (エンジン回転数)		スロットル	クーラン ト温度	フラグ	使用せず	使用せず	使用せず

エンジン制御器はこのようなフレームを定期的送信します。この例では、ID432 の最初の 2 データ バイト(1 バイトは 8 ビットなので 16 ビット。) は、RPM を 表すために使用されます。3 番目のバイト は、スロットルポジションセンサーの値を含んでいます。4 番目のバイト は、エンジンクーラント温度です。5 番目の バイト は、スイッチ入力等、バイナリ情報のためのフラグを含んでいます。この例では、6、7、8 番目の バイト は使用されません。例えばこの例において CAN バス に接続しているある制御ユニットが RPM 情報 を必要としている場合は、制御ユニットに対して ID 432 を含むメッセージが伝達され、フレーム内 8 バイトのうち、最初の 2 バイト が制御ユニット側により抽出されればよいことになります。

「標準」CAN ID に対して「拡張」CAN ID も存在します。標準 ID との違いは、ID に使用しているビット数で標準 CAN ID が 11 ビットなのに対して拡張 CAN ID は 29 ビット を使用しています。これによって信号に割り当てることができる ID の数が非常に多くなります。拡張 CAN ID は例えばトラック用の J1939 基準で使用しています。CAN02 は、デフォルトで自動的に拡張タイプ ID および標準タイプ ID を受信します。

CAN02 と VBOX との接続



付属の CAN ケーブル（RLCAB006 もしくは RLCAB005-C）を使い、VBOX の CAN-Bus ソケットと CAN02 のソケット 1（CAN / POWER IN）またはソケット 2（CAN OUT）に接続します。

VBOX に電源を供給します。

VBOX と PC を USB ケーブルで接続します。

※PC との接続時、COM ポートが異なっている場合は CAN02 が認識されません。その場合は「コントロールパネル」→「デバイスマネージャ」で COM ポートを確認し、再度接続して下さい。

セットアップ

CAN02 のセットアップ

CAN02 を使用する前にはセットアップが必要です。セットアップにより、CAN-Bus 内のどの ID からデータを読み込むのかを認識することができます。CAN02 を設定するには、VBOXTools ソフトウェアのセットアップ機能を使用します。

CAN02 を VBOX に接続し、VBOX に接続されている PC から VBOX Tools ソフトウェアを実行し、[VBOX Setup] をクリックします。CAN02 が接続されていると、VBOX は自動的に CAN02 ユニットを認識し、セットアップ画面の [Standard Channels] タブの横に [VCI Modules] タブとして表示されます。[VCI Modules] タブ内にはさらにシリアル No.のタブが表示されており、セットアップするユニットをクリックし、チャンネルボタンを表示させます。

コンパクトフラッシュにログをしたいチャンネルのチェックボックスにチェックを入れます。

チャンネルボタンをクリックすることで、各チャンネルのセットアップが編集できます。

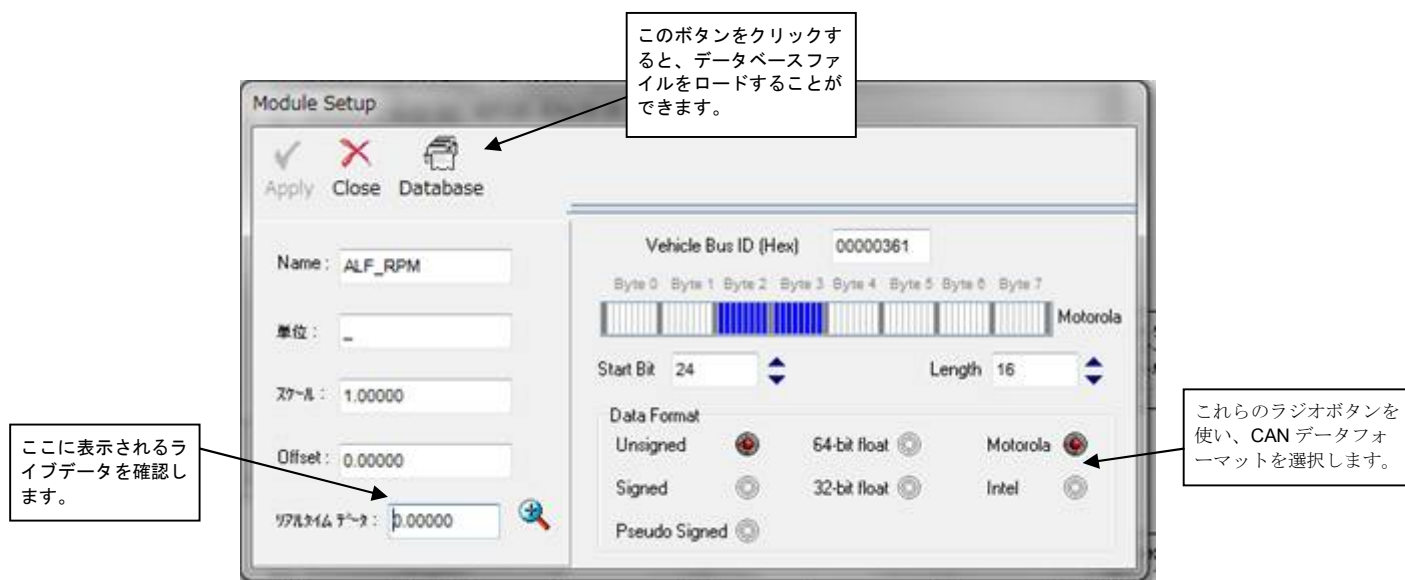
ボーレートの設定はこのボタンをクリックします

タブにはシリアルナンバーが記載されています。設定したい CAN02 であるかどうかを確認し、設定して下さい。

各チャンネルの[log to compact flash]のチェックボックスにチェックを入れ、VBOX がチャンネルを記録できる状態にします。各チャンネルの詳細を編集するには、目的のチャンネル名ボタンをクリックし、[Module Setup] ウィンドウを表示させます。

下の画面が、[Module Setup] ウィンドウです。このウィンドウで、チャンネル名、チャンネル単位、スケール、オフセット、および CAN パラメータを編集できます。

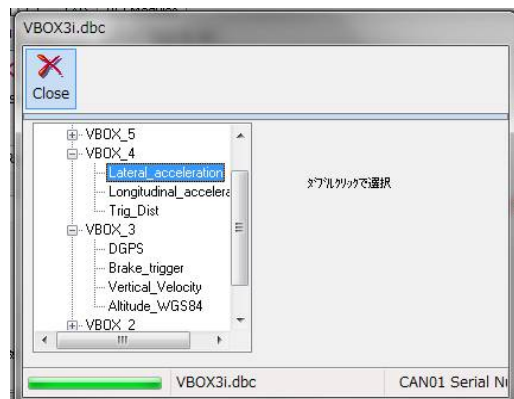
チャンネルをセットアップした後に、[Module Setup] ウィンドウの [リアルタイムデータ] 欄を確認します。この欄を確認すると、データおよびスケールが正確かどうか速やかに分かります。



Database ボタンを利用すると、CAN データベースファイル（拡張子“DBC”のファイル）を読み込むことや、Racelogic 社が独自に解析をした拡張子“REF”のファイルを読み込むことも可能です。REF ファイルには、Racelogic 社が解析した CAN データ信号が含まれおり、多くの自動車メーカーの基本機能を提供します。データセットの多くは、少なくとも以下の信号パラメータを含んでいます。

車両速度 / 車輪速度 – 通常は各ホイールの速度
 エンジン回転数
 ブレーキスイッチ
 スロットルポジション

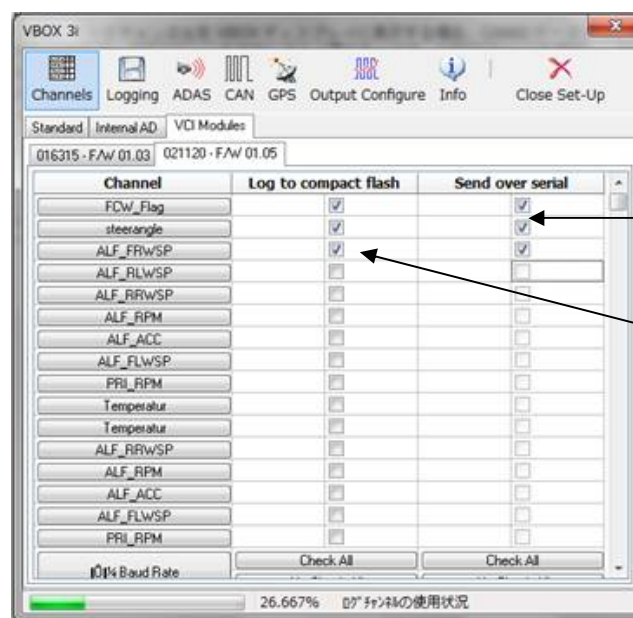
データベースファイルから設定するには、[Database File] をクリックします。その後、Windows の標準ブラウザウィンドウが表示されるので、そのウィンドウから必要なデータベースを選択しロードします。次に、ウィンドウが表示されますので、このウィンドウから利用したい CAN データを選択します。



シリアルデータ

CAN02 データチャンネルを VBOX の各 ディスプレイに表示する場合、CAN02 データチャンネルのシリアルデータ通信を有効にする必要があります。

チャンネル名の横の [Send over serial] にチェックを入れます。

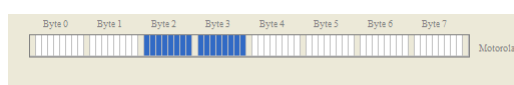


ボックスにチェックを入れ、ディスプレイ表示したいチャンネルのシリアルデータストリームを有効にします。(コンパクトフラッシュへの記録を有効にしないと、シリアルデータの送信は有効になりません。)

信号パラメータの解説

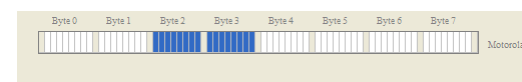
信号名 データパラメータの解説

スタート ビット データフレーム内にあるデータの一番先頭のビットの位置を数字で表したものです。CAN データフレーム内の信号位置は、バイト の位置ではなく ビット の位置により照合されます。従って、ビット の位置は 0～63 の値になります。ビット 0 の位置は、Motorola または Intel フォーマットボタンのどちらを有効にするかにより変化します。ビット のナンバリングに関しては、「**CAN で使用するデータフォーマット**」のセクションで詳細を確認してください。

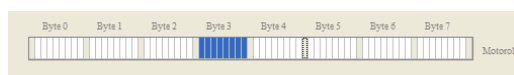


信号ディスプレイ上の青いハイライト表示は、8 バイトの CAN フレームのどの ビット から信号が由来しているのかを示しています。スタート ビット の値が変わると、青いハイライト表示の位置も変わります。

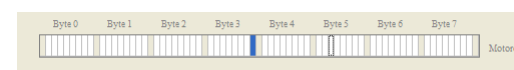
信号長 信号長を ビット で表します。信号長は 1～16 ビット の範囲です。例えば、ブレーキスイッチフラグ(on/off)の信号長は、1 ビット になります。スロットルポジション信号長は 8 ビット (1 バイト)、車輪速度は一般的に 16 ビット が多いです。



16 ビット 信号



8 ビット 信号



1 ビット 信号

Signed / Unsigned CAN02 が CAN メッセージから信号データを抽出した後、CAN02 はそのデータを Signed(符号付き) /Unsigned(符号なし)のどちらで扱うべきか認識する必要があります。Signed では、正の数も負の数も表示可能です。一方 Unsigned で符号の表示ができなくなります。Unsigned の例は、0～8000 RPM のエンジン回転数です。またステア角の信号は正または負の値を含んでおり、そのために Signed の値として表されます。

Intel / Motorola	いずれかを選択することで、CAN フレーム内で信号データがどのように表されるか、CAN02 に指示します。「 CAN で使用するデータフォーマット 」のセクションで、Intel と Motorola のフォーマッティングの違いを説明しています。
スケールおよびオフセット	CAN バス からの信号値は、時として生データであり、実際の単位に直接結びつけられないことがあります。スケールおよびオフセット機能を用いることによって、実データを記録することができます。

CAN で使用するデータフォーマット

CAN メッセージでデータを表す方法を更に複雑化するために、自動車メーカーは主に 2 つの方法を用いて、メッセージ内のビットの順序やバイトの順序を表します。

例 1 低位 ←———— バイト —————→ 高位

ID	バイト 1	バイト 2	バイト 3	バイト 4	バイト 5	バイト 6	バイト 7	バイト 8
298	上位 バ イト	下位バ イト						

例 2 低位 ←———— バイト —————→ 高位

ID	バイト 1	バイト 2	バイト 3	バイト 4	バイト 5	バイト 6	バイト 7	バイト 8
298	下位 バ イト	上位 バ イト						

これらの例は、ID が同一の 2 種類の CAN メッセージです。いずれの例でも、CAN フレームのバイト 1 と 2 に 16 ビットが格納されています。例 1 では 16 ビット(0-15 ビット)のうち、上位 バイト(最高位のビットが含まれるバイト) がフレームの位置 1、下位 バイト(最低位のビットが含まれるバイト) が位置 2 に格納されています。例 2 では、16 ビットのうちバイト 1 に下位 バイト、続いて 2 に上位 バイト が格納されています。

このように、データを格納するには 2 種類の方法がありますが、これらはそれぞれ Intel タイプ、および Motorola タイプと呼ばれます。

Intel は、バイト 順として「リトルエンディアン」を使用します。

例えば 2 バイト(16 ビット)が使用される場合「リトルエンディアン」では、LSB(Least Significant Byte 最下位バイト:0-7 ビット) は使用されるメモリの最下位部分に、MSB(Most Significant Byte 最上位バイト 8-15 ビット) は最高位部分に格納されます。

Motorola は、バイト 順として「ビッグエンディアン」を使用します。

「ビッグエンディアン」では、MSB は使用されるメモリの最下位部分、LSB は最高位部分に格納されます。

例えばスタートビット(ナンバー)が「0」、32 ビット長(4 バイト数)の場合、Intel フォーマットではバイト 1→バイト 4 の順で格納され、Motorola フォーマットを用いた場合にはバイト 8→バイト 1 の順で格納されます。

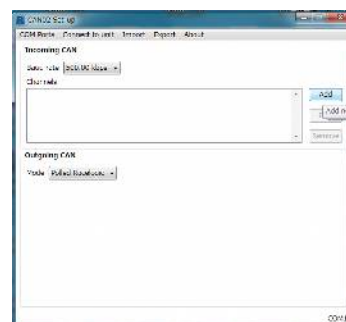
Intel フォーマットで格納される CAN データの ビット ナンバリング								
バイト No.	1	2	3	4	5	6	7	8
ビット No.	7.....0	15.....8	23.....16	31.....24	39.....32	47.....40	55.....48	63.....49

Motorola フォーマットで格納される CAN データの ビット ナンバリング								
バイト No.	1	2	3	4	5	6	7	8
ビット No.	63.....56	55.....48	47.....40	39.....32	31.....24	23.....16	15.....8	7.....0

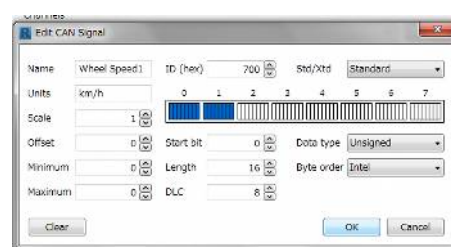
CAN フレームで Intel フォーマットを使う自動車メーカーもあれば、Motorola フォーマットを使う自動車メーカーもあります。従って、例えばエンジン回転数の 16 ビット 信号は、上位 バイト が先で下位 バイト が後、または下位 バイト が先で上位 バイト が後、の順で CAN フレームに格納されます。どちらになるかは自動車メーカーにより異なります。

CAN データベースの作成

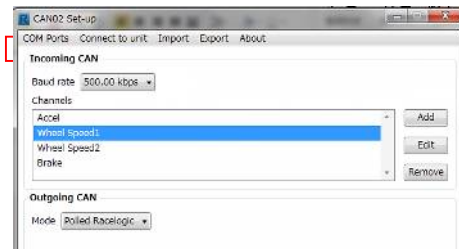
独自の CAN データベースを新たに作成するには、CAN02Set-up ソフトウェアを使用します。CAN02 を VBOX に接続して電源をとり、RLVBCAB01 ケーブルを用いて PC と接続します。画面が表示されましたら右側の[Add]ボタンをクリックします。



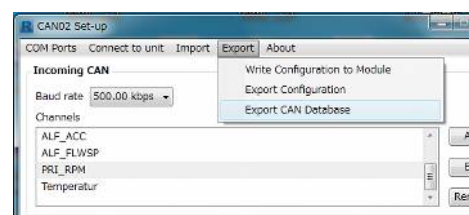
右図が表示されますので、CAN の設定をしていきます。ID やバイトオーダー(Start bit や Length)、ユニット等、全ての情報の入力を終えて[OK ボタン]をクリックすると新しいチャンネルが追加されます。複数のチャンネルを作成する場合は[Add]ボタンでチャンネルを追加して下さい。



CAN 情報の設定を変更したい場合には、チャンネルを選択後に[Edit]ボタンを押して入力画面に進んでください。不要なチャンネルを削除したい場合は削除したいチャンネルを選択し、[Remove]ボタンを押して下さい。ボーレートを変更したい場合は[Baud rate]のプルダウンから希望したいレートを選択して下さい。



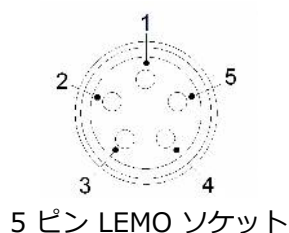
全ての設定が終わったら、上部の[Export]タブでファイル名をつけて保存します。ID やスタートビット、ビット長等、設定内容を含めて保存したい場合は[Export Configuration] で保存して下さい。



拡張子が DBC (CAN Database) もしくは RMSF (Configuration) とついた CAN データベースが出来上がります。



ピン配列



コネクタ 1 – CAN / 電力入力

ピン	I/O	機能	
1	I/O	コネクタ 2 ピン 1 への直接接続	
2	I/O	コネクタ 2 ピン 2 への直接接続	
3	I/O	CAN High	VBOX CAN BUS
4	I/O	CAN Low	VBOX CAN BUS
5	O	+12 V 電源	
シールド		GND	

コネクタ 2 – CAN / 電力出力

ピン	I/O	機能	
1	I/O	コネクタ 1 ピン 1 への直接接続	
2	I/O	コネクタ 1 ピン 2 への直接接続	
3	I/O	CAN High	VBOX CAN BUS
4	I/O	CAN Low	VBOX CAN BUS
5	O	+12 V 電源	
シールド		GND	

コネクタ 3 - ファームウェアアップグレード

ピン	I/O	機能	
1	O	TxD、シリアルデータ送信	ファームウェアのアップグレードに使用
2	I	RxD、シリアルデータ受信	ファームウェアのアップグレードに使用
3	-		
4	-		
5	O	+12 V 電源	
シールド		GND	

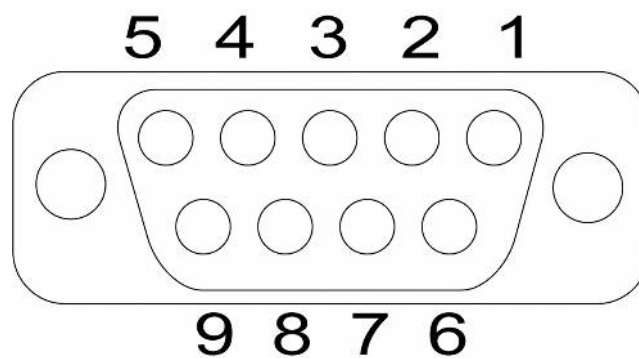
コネクタ 4 - 車両 CAN バス

ピン	I/O	機能	
1	-		
2	-		
3	I/O	CAN High	車両 CAN バス
4	I/O	CAN Low	車両 CAN バス
5	I	+12 V 電源	
シールド		GND	

LEMO から 9 ピン D への RLVBCAB19 ケーブル

9 ピン D サブコネクタ

ピン	機能	LEMO ピンに接続
1		
2	CAN BUS Low	4
3		
4		
5	CAN BUS シールド	ケース
6		
7	CAN BUS High	3
8		
9	+12 V	5



モジュール寸法

