



# CANalyzer

プロダクトインフォメーション

## 目次

|        |                       |    |
|--------|-----------------------|----|
| 1      | CANalyzer の概要 .....   | 4  |
| 1.1    | 特長／優位性 .....          | 4  |
| 1.2    | 適用分野 .....            | 5  |
| 1.3    | バスシステムおよびプロトコル .....  | 5  |
| 1.4    | CANalyzer のグレード ..... | 5  |
| 1.5    | システム要件 .....          | 6  |
| 1.6    | 詳細情報 .....            | 6  |
| 2      | 機能 .....              | 6  |
| 2.1    | 特殊機能 .....            | 7  |
| 2.2    | データベースのサポート .....     | 7  |
| 3      | 解析と刺激入力 .....         | 7  |
| 3.1    | 解析 Window .....       | 7  |
| 3.1.1  | 測定設定 .....            | 8  |
| 3.1.2  | トレース Window .....     | 9  |
| 3.1.3  | グラフィック Window .....   | 10 |
| 3.1.4  | スコープ Window .....     | 11 |
| 3.1.5  | データ Window .....      | 12 |
| 3.1.6  | 統計 Window .....       | 12 |
| 3.1.7  | 出力 Window .....       | 13 |
| 3.1.8  | マップ Window .....      | 14 |
| 3.1.9  | インタラクティブジェネレーター ..... | 15 |
| 3.1.10 | トリガーとフィルター .....      | 16 |
| 3.1.11 | ロギング／リプレイ .....       | 16 |
| 3.2    | データの解釈 .....          | 17 |
| 3.3    | オフライン評価 .....         | 17 |
| 3.4    | エクスポート機能 .....        | 17 |
| 3.5    | 統合デスクトップ .....        | 17 |
| 3.6    | 開始値の設定 .....          | 18 |
| 3.7    | シンボルマッピング .....       | 18 |
| 4      | 診断 .....              | 19 |
| 5      | プログラミング .....         | 23 |
| 5.1    | CAPL インターフェイス .....   | 23 |
| 5.1.1  | C ライクな構文 .....        | 23 |
| 5.1.2  | イベント指向制御 .....        | 23 |
| 5.1.3  | シンボリックアクセス .....      | 24 |
| 5.1.4  | アプリケーション固有の言語拡張 ..... | 24 |
| 5.1.5  | CAPL ブラウザー .....      | 25 |
| 5.2    | ビジュアルシーケンサー .....     | 26 |
| 6      | パネル .....             | 26 |
| 7      | ソフトウェアインターフェイス .....  | 27 |
| 8      | ネットワークインターフェイス .....  | 27 |
| 9      | オプション Scope .....     | 28 |
| 9.1    | 適用分野 .....            | 28 |

|       |                         |    |
|-------|-------------------------|----|
| 9.2   | 主な特長.....               | 28 |
| 9.3   | サポートするプロトコル .....       | 28 |
| 9.4   | 対応するオシロスコープハードウェア ..... | 29 |
| 9.5   | オシロスコープソフトウェア .....     | 29 |
| 9.5.1 | 設定機能 .....              | 29 |
| 9.5.2 | トリガー機能.....             | 29 |
| 9.5.3 | 解析機能 .....              | 30 |
| 9.5.4 | オフライン機能 .....           | 30 |
| 10    | トレーニング .....            | 30 |

V6.3 06/2020

CANalyzer バージョン 13.0 以降を対象としています。

この資料では、解析、刺激入力、診断といった CANalyzer の適用分野と、それぞれの機能について説明します。また、CANalyzer でのプログラミング、追加プログラム、ネットワークインターフェイスについても簡単に概説します。

CANalyzer FlexRay、LIN、MOST、CANopen オプションについては、別途「プロダクトインフォメーション」をご用意しております。

発行元：ベクター・ジャパン株式会社

[www.vector.com/jp/ja/](http://www.vector.com/jp/ja/)

※記述されている内容は予告なく変更されることがあります。(発行日:2020 年 6 月 26 日)

## 1 CANalyzer の概要

CANalyzer は、ECU ネットワークおよび分散システムのための汎用解析ツールです。モニターや解析に加え、サポートされているバスシステムへ容易にデータを送信することもできます。

ツールの強力な基本機能とユーザープログラミング性により、簡単なネットワーク解析から複雑な問題に対するデバッグまで、あらゆるニーズに対応します。

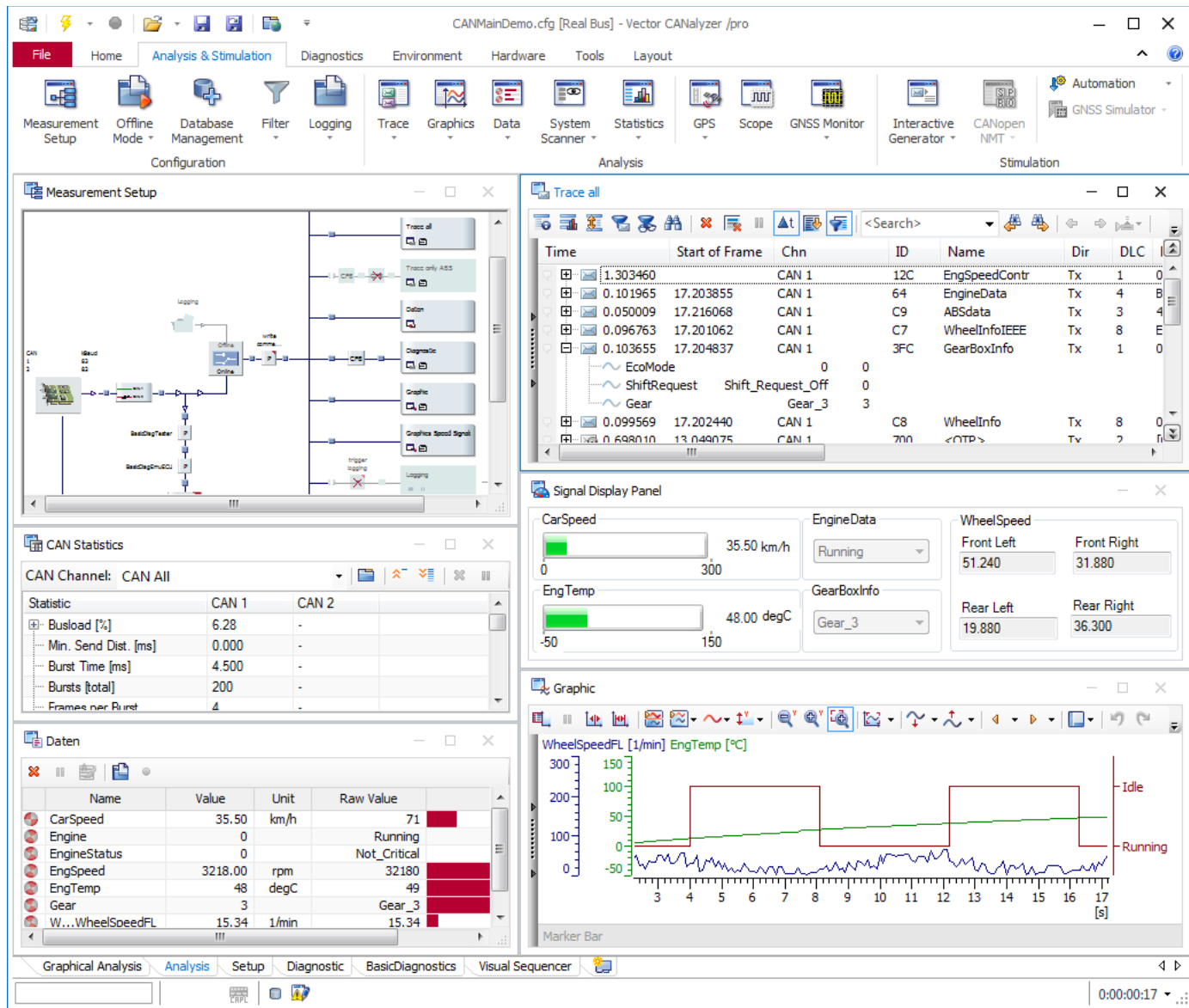


図 1: CAN システムの解析用の CANalyzer による標準的なコンフィギュレーション

### 1.1 特長／優位性

CANalyzer ではグラフィックなブロックダイアグラムから直観的に操作できます。PC インターフェイスを介してバスのデータの流れをさまざまな評価 Window に表示し、ログファイルに記録します。システムは、このブロックダイアグラム内でパラメーター化されます。フィルター、ジェネレーター、リプレイブロックなどの機能ブロックも、このブロックに挿入して設定できます。

## 1.2 適用分野

CANalyzer は、あらゆる用途にお使いいただけます。簡単なネットワーク解析から、高度な解析、エミュレーションシステムまで、あらゆるレベルの用途に最適です。たとえば、リプレイブロックを送信ブランチに挿入し、バス上で以前に記録したバスシステムのデータトラフィックを再生することができます。また、出力フィルターを追加して、データトラフィックの一部を送信しないこともできます。これらにより、ECU テストにとって理想的なテスト環境を実現することができます。

CANalyzer のその他のユースケース例:

- > バスモジュールのエミュレーション
- > 簡単な残りのバスシミュレーションやテストシーケンスの作成
- > メッセージ操作機能を利用可能な 2 つのバス間のゲートウェイ
- > プログラムを使用したトリガーによるクリティカルフェーズのログ記録
- > テキストメッセージの表示によるユーザー固有のオンライン評価
- > 簡単な ECU 診断機能の確認

## 1.3 バスシステムおよびプロトコル

CANalyzer ではさまざまなオプションが提供されており、さまざまなバスシステム、CAN ベースプロトコルと任意に組み合わせて使用できます。

CANalyzer がサポートする**バスシステム**: CAN、CAN FD、LIN、MOST、FlexRay、J1708、Ethernet、K-Line、A429、WLAN、AFDX<sup>1</sup>

**CAN ベースの上位プロトコル**: J1939、CANopen、CANaero

これ以外のプロトコルについてはご相談ください。

オプションの詳細については、ベクターの Web サイトに掲載されています。

## 1.4 CANalyzer のグレード

- > **CANalyzer fun**: fun (Fundamental) 版は、簡単な解析タスクに最適で、インタラクティブな標準機能はすべて利用できます。ただし、プログラミング機能や診断テスター、制御パネルは利用できません。
- > **CANalyzer exp**: exp (Expert) 版は、ほぼどのような標準的用途にも適しており、CAPL プログラムの作成および実行以外のすべての機能および拡張機能を無制限に利用できます。
- > **CANalyzer pro**: pro (Professional) 版では、すべての機能および拡張機能を無制限に利用できます。このグレードは、バストラフィックの簡単なモニターから、マルチバスシステムの複雑な解析や刺激入力、テストまで、すべての用途に対応しています。

---

<sup>1</sup> AFDX®は Airbus の登録商標です。

## 1.5 システム要件

| コンポーネント       | 推奨                                                                                                                      | 最小要件                                                                                                                                                           |
|---------------|-------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CPU           | <ul style="list-style-type: none"> <li>&gt; インテル Core i7 または互換 CPU</li> <li>&gt; ≥ 3GHz</li> <li>&gt; ≥ 4 コア</li> </ul> | <ul style="list-style-type: none"> <li>&gt; インテル互換</li> <li>&gt; 2GHz</li> <li>&gt; 2 コア</li> </ul>                                                            |
|               | CANalyzer の性能は CPU のクロック周波数に大きな影響を受けます。                                                                                 |                                                                                                                                                                |
| メモリー (RAM)    | ≥ 32GB                                                                                                                  | 8GB                                                                                                                                                            |
| ハードディスク容量     | ≥ 20GB SSD/NVMe                                                                                                         | 8GB HDD/SSD                                                                                                                                                    |
|               | (使用するオプションと、オペレーティングシステムコンポーネントにより異なる)                                                                                  |                                                                                                                                                                |
| 画面解像度         | フル HD                                                                                                                   | 1280 × 1024 ピクセル                                                                                                                                               |
| オペレーティングシステム* | <ul style="list-style-type: none"> <li>&gt; Windows 10 64bit (バージョン 1803 以上)</li> </ul>                                 | <ul style="list-style-type: none"> <li>&gt; Windows 10 64bit (バージョン 1803 以上)</li> <li>&gt; Windows 8.1 64bit</li> <li>&gt; Windows 7 64bit (SP1 以上)</li> </ul> |
|               | * 仮想 OS を除く。仮想マシン上での実行は可能ですが、テストは行われておりません。仮想化により、ベクター製ハードウェアの動作に遅延時間が大きくなるなどの影響が生じるおそれがあります。                           |                                                                                                                                                                |

## 1.6 詳細情報

### > サポート/ダウンロード

### > デモバージョン

CANalyzer の各種デモバージョンをベクターの Web サイトよりダウンロードできます。デモバージョンでは、各種適用分野のサンプルコンフィギュレーションおよび CANalyzer のすべての機能についてのヘルプをご利用いただけます。

### > CANalyzer の機能マトリクス

CANalyzer のグレード、チャンネル、サポートされるバスシステムの詳細については、別途、データシート「CANoe/CANalyzer の機能マトリクス」をご参照ください。

## 2 機能

CANalyzer の基本機能により、多種多様な用途に対応しています。基本機能は、次のとおりです。

- > バスデータトラフィックのトレース表示
- > シグナル値のグラフィックおよびテキスト表示
- > 定義済みメッセージのインタラクティブ送信
- > ログファイルに記録されたメッセージの送信
- > KWP2000 および UDS 準拠の診断通信。本格的な診断テスターとして使用可能
- > メッセージに関する統計
- > バス負荷およびバス障害に関する統計
- > メッセージのログ記録をリプレイまたはオフライン環境での評価
- > バス障害の生成
- > 柔軟なドッキングコンセプトとユーザーフレンドリーなメニュー構造を採用した直観的なインターフェイス
- > CANalyzer をサポートするベクター製の新ハードウェア：
  - > VN1610 (2 チャンネル – CAN)
  - > VN1611 (2 チャンネル – CAN および LIN/K-Line)

- > VN1630A (4 チャンネル – CAN および LIN/K-Line)
- > VN1640A (4 チャンネル – CAN および LIN/K-Line)

## 2.1 特殊機能

CANalyzer の機能の主な特長は、次のとおりです。

- > 簡単かつ直観的なユーザー制御コンセプト
- > ビジュアルシーケンサーを使用して、刺激入力およびテストシーケンスを対話形式で作成
- > CAPL を使用してユーザープログラミングが可能
- > バス固有の各種機能ブロックを利用可能
- > 診断機能:
  - > ODX 2.0.1/2.2.0、MDX 2.0/3.0、CDD 形式の診断データベースファイルにより診断をパラメーター化
  - > ベーシックダイアグノスティックエディターを使用して簡単な診断サービスを定義
  - > フィジカルアドレッシングとファンクショナルアドレッシングをサポート
  - > 搭載されている OBD-II テスターを使用して、オンボード診断を迅速かつ簡単に実行可能
  - > パラメーター化可能な診断データベースに基づく UDS および KWP2000 用の診断オブザーバー
  - > ISO/DIS 15765-2 用のトランスポートプロトコルオブザーバー
  - > DoIP (Diagnostics over IP) および HSFZ (High-Speed-Fahrzeug-Zugang) のサポート

## 2.2 データベースのサポート

CANalyzer は、次の形式に基づくデータベースをサポートします: DBC (CAN)、LDF (LIN)、XML (MOST)、FIBEX (FlexRay)、AUTOSAR システムディスクリプション (CAN/FlexRay/Ethernet)。

CANalyzer は、次の診断データベースファイルをサポートします: CDD (CANdelaStudio)、ODX 2.0.1/2.2.0 (PDX ファイル)、および MDX 2.0/3.0。

CANalyzer では、これらデータベースからの情報をシンボルとして表示し、CAPL などで使用できます。

## 3 解析と刺激入力

CANalyzer による解析の基本は、データソースからその表示やロギングまでのデータフローです。たとえば、フィルターを統合して、解析時に考慮しなければならないデータと、そうでないデータを定義するなどのデータ処理も可能です。また、さまざまなステイミュレーションオプションで、データトラフィックを確認することができます。

主な特長

- > ドラッグ & ドロップによる容易な解析 Window 設定。メッセージやシグナルを、ある解析 Window から別の解析 Window へコピーまたは移動することが可能
- > 複数の機能を解析する場合、1 つのタイプの Window (グラフィック Window など) をデータフローに複数配置して、同時に解析が可能
- > ステータスバーからロギングを直接、マウスクリックで簡単に開始および停止可能

### 3.1 解析 Window

CANalyzer には、以下のような Window およびブロックがあります。

### 3.1.1 測定設定

測定設定では、データフローをグラフィック表示して設定します。

- > **データソースの定義**(オンライン/オフライン)  
オンラインデータソースとして、ハードウェア(CANcardXL など)によって接続される実バスまたはシミュレーションバスを使用します。ログデータが保存されたファイルをオフラインデータソースとして使用します。
- > **解析 Window 挿入**  
解析要件に応じて、データを個々の Window に各種形式で表示できます。たとえば、シグナル波形をグラフィック表示したり、シグナル値を表示したりできます。
- > **CAPL プログラムノードの挿入**  
データのフィルタリングや各種算術演算などのタスクに、CAPL プログラムノードを使用できます。
- > **フィルターの挿入**  
フィルターを使用することでデータをよりわかりやすく表示できます。フィルターで、どのデータを渡し、どのデータをブロックすべきかを定義します。フィルターは、測定中または測定後にアクティブにでき、個々のシグナルからバスシステム全体のチャンネルまで、幅広いオブジェクトをフィルターできます。
- > **トリガー条件の挿入**  
フィルターと同様に、トリガー条件を利用してデータ量を減らすこともできます。トリガーは、特にバスイベントに反応するように設定され、互いに組み合わせることができます。
- > **データの記録**  
測定後の解析のために、データをログファイルに記録できます。ログファイルは、後でオフラインデータソースとして再利用したり、リプレイしたりできます。

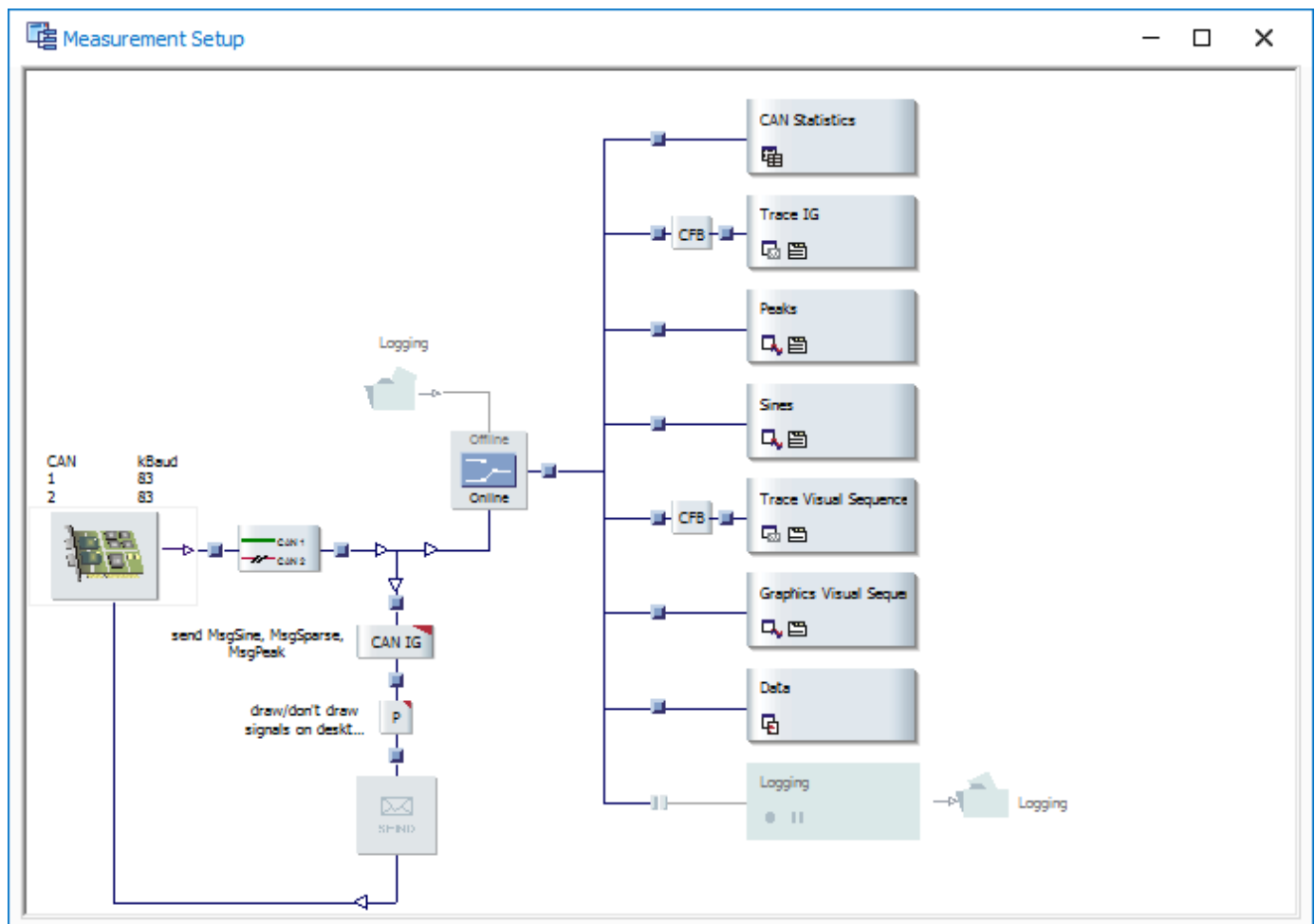


図 2: オンラインデータソースを用いた測定設定

### 3.1.2 トレース Window

トレース Window には、メッセージやエラーフレームの送信などのバスアクティビティがリスト表示されます。各メッセージの個々のシグナル値を表示できます。下記のような機能を使用してデータを解析できます。

#### > フィルターの挿入

トレース Window では、各種フィルターを使用できます。フィルターを使用して、表示するデータ量を減らしたり、データをデータストリームから削除することもできます。

#### > 未変化データの非表示

より見やすくするために、変化のないデータについては色を徐々に薄くしたり、画面で非表示にすることができます。

#### > イベントのハイライト表示

重要なイベントとメッセージに色をつけて強調表示できます。

#### > マーカーの設定

マーカーを設定すると、イベントを識別して迅速に見つけることができます。マーカーは 1 つのイベントに割り当てられるため、そのタイムスタンプにも割り当てられます。設定したマーカーは、他の解析 Window に表示することもできます。

#### > 統計の表示

値を含むメッセージ/シグナルをさまざまな観点から各種ビューで詳細に表示します。タイムスタンプ間やシグナル値間の差異も計算されます。

#### > データの記録

トレース Window の内容の一部またはすべてをエクスポートできます。その後、エクスポート済みのファイルを別のフォーマットに変換し、同じデータセットを別のプログラムでさらに処理することもできます。

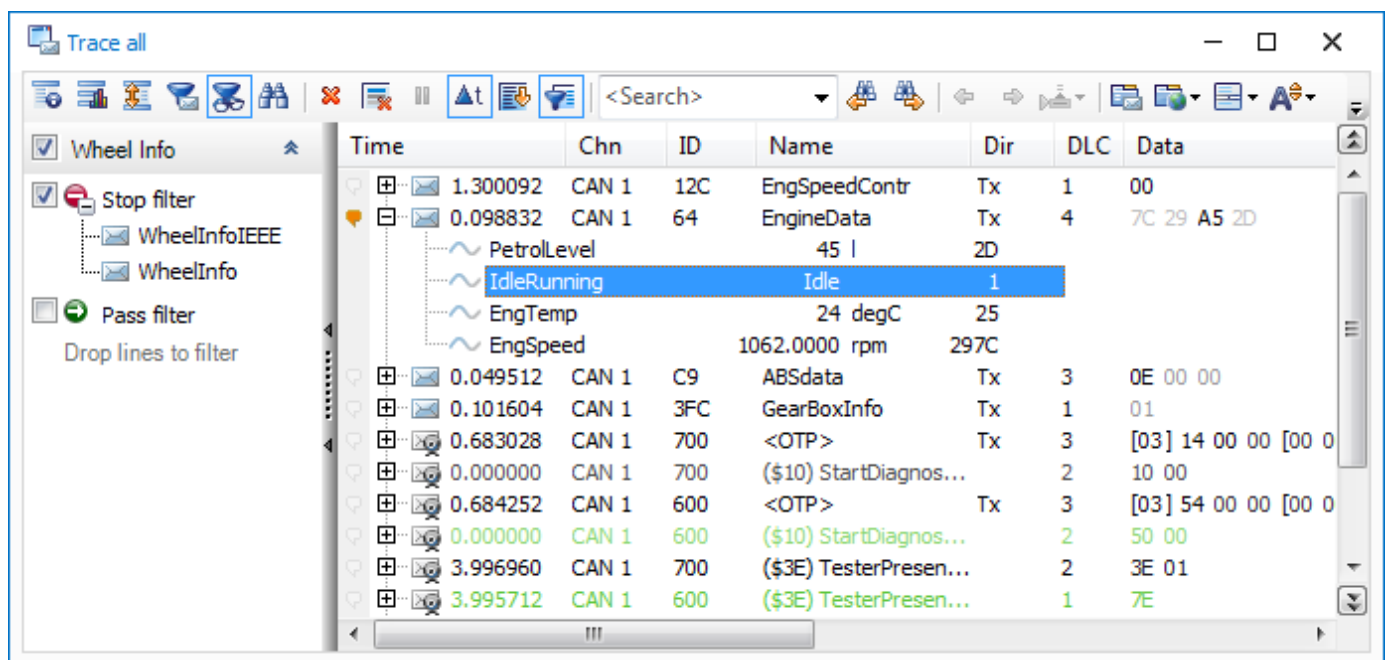


図 3: ブロックフィルターがアクティブでマーカーが設定されているトレース Window

### 3.1.3 グラフィック Window

グラフィック Window を使用して、シグナル、環境データ、診断パラメーターの値をカーブとしてグラフィック表示します。これらのカーブの測定と評価に使用できる機能のいくつかを以下に示します。

#### > 測定マーカー／差異マーカーの表示

測定マーカーまたは差異マーカーを使用して、測定値の絶対的または相対的な解析を行うことができます。  
測定マーカーは、トレース Window の表示と同期させることができます。

#### > マーカーの設定

マーカーを設定すると、イベントを識別して迅速に見つけることができます。マーカーは 1 つのイベントに割り当てられるため、そのタイムスタンプにも割り当てられます。設定したマーカーは、他の解析 Window に表示することもできます。

#### > 測定カラムの表示

凡例部分に、各シグナルの全体または差異選択部分の最小値と最大値を表示したり、同じタイプのシグナル間の Y 軸の差異を表示できます。

#### > 統計の表示

最小値、最大値、平均値、標準偏差などの統計データは、グラフィック Window で選択されたシグナルまたはすべてのシグナルに適用できます。

#### > X/Y モード

シグナルリストを右クリックすると、選択したシグナルを X 軸に設定することもできます。

#### > データの記録

グラフィック Window のシグナルを測定時に自動または手動で記録できます。このとき、メッセージからシグナルを抽出して、バイナリー形式でシグナルベースの MDF ファイルに保存します。

グラフィック Window のシグナル波形すべて、またはシグナル波形の表示セクションのみをファイルに保存できます。

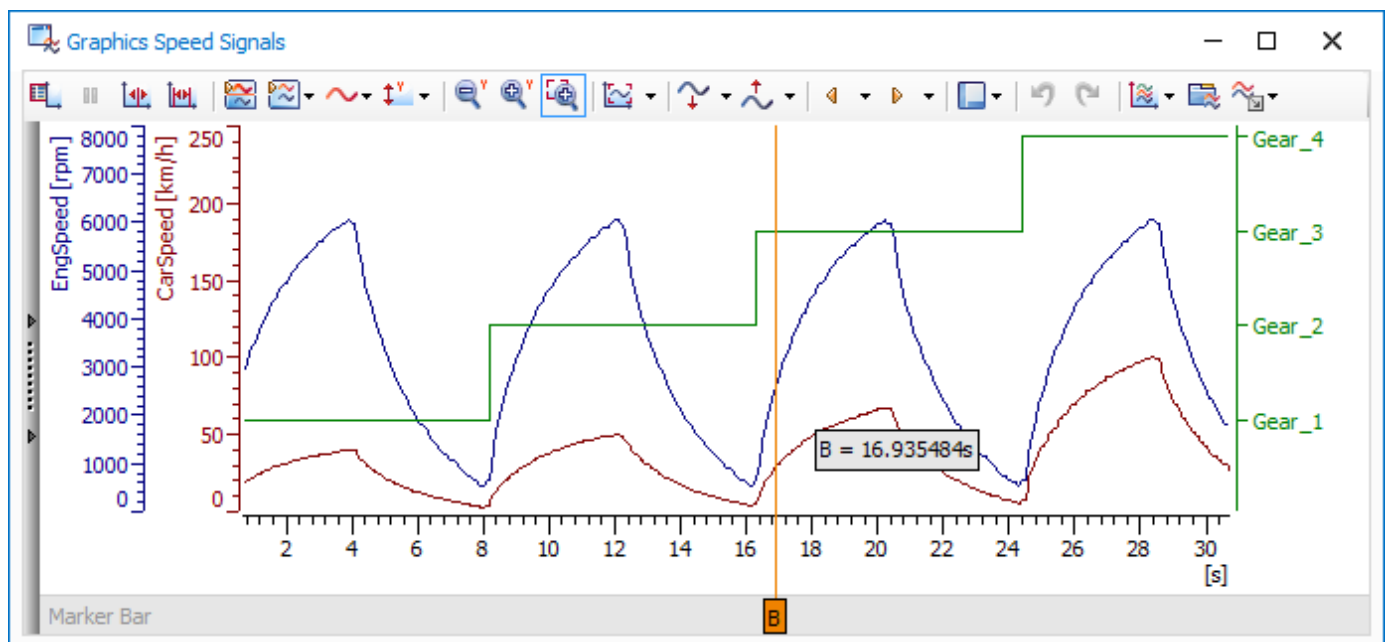


図 4: マーカーが設定されたグラフィック Window

### 3.1.4 スコープ Window

スコープ Window は、バスレベルの測定をグラフィック表示し、プロトコルエラーの解析に使用されます(第 9 章の「オプション Scope」もご参照ください)。

#### > トリガーの設定

スコープ Window では、CAPL あるいはあらかじめ設定されている条件を用いて手動でトリガーを行うことができます。任意の数のトリガー条件を作成でき、個々のトリガー条件は論理和関係で組み合わせることができます。

#### > 測定値の解析

ダイアグラムに測定値とそのデコード内容がグラフィック表示されます。

#### > シグナルの比較

さまざまなアプローチでデータを比較できます。たとえば、同じデータを異なる時間ベースで取得した場合のデータや、異なるデータを同じ時間ベースで比較できます。

#### > データの記録

取得したデータをエクスポートして、後で解析用にインポートできます。

#### > 測定カーソル

測定カーソルは時間や電圧などの物理的なデータだけでなく、それらの値を他のカーソル位置と比較し、その差を測定するのに役立ちます。カーソルのツールチップにはビットの論理値(例:ドミナント/レセッシブ)が表示されます。すべての物理値は独立した凡例リストに表示されます。

#### > グローバルマーカー

スコープ Window には、測定の重要ポイントをマークするためのマーカーを設定できます。それぞれのマーカーには名前と、正確なタイムスタンプが含まれます。

#### > アイダイアグラム

ビットの重畳または理想的なビットの表示など、拡張された解析が可能です。

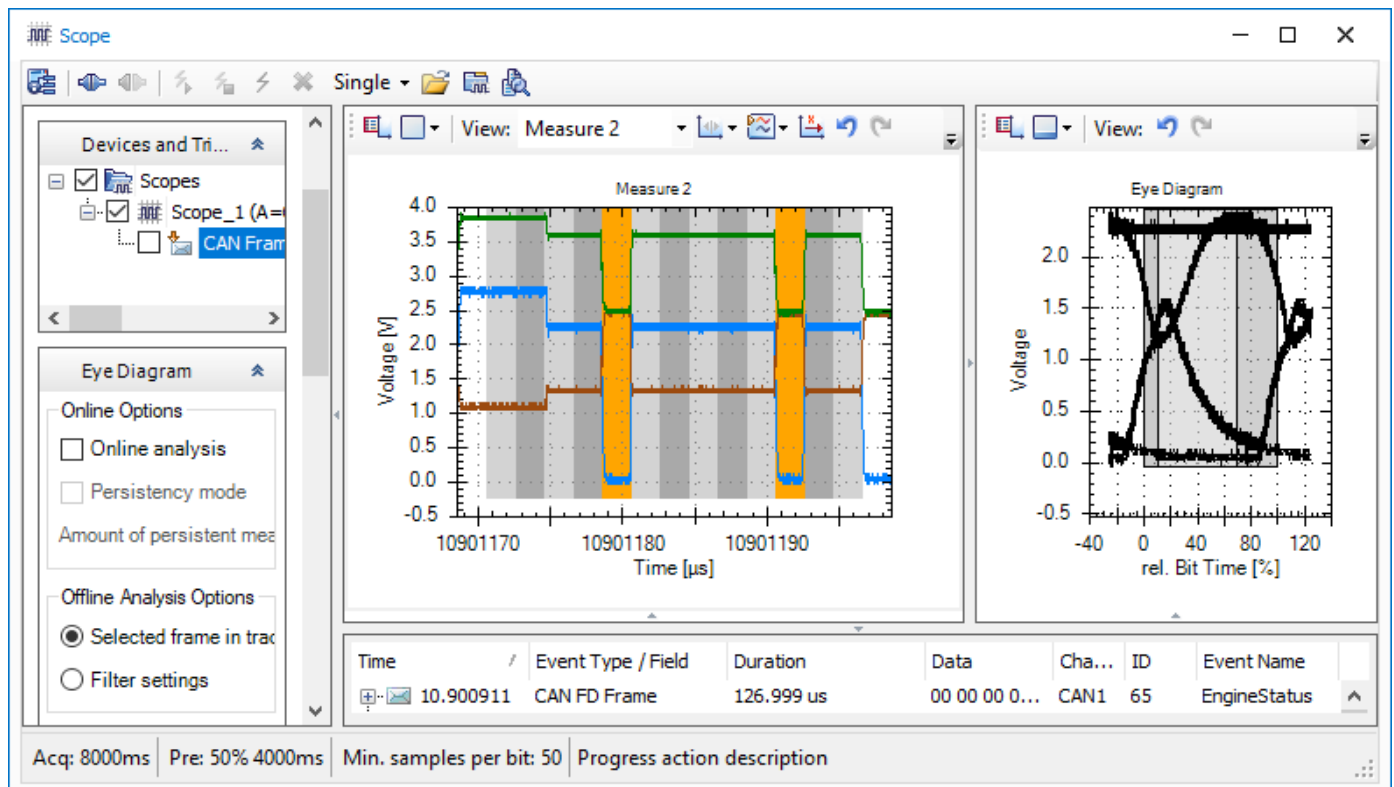


図 5: アイダイアグラムが表示されたスコープ Window

### 3.1.5 データ Window

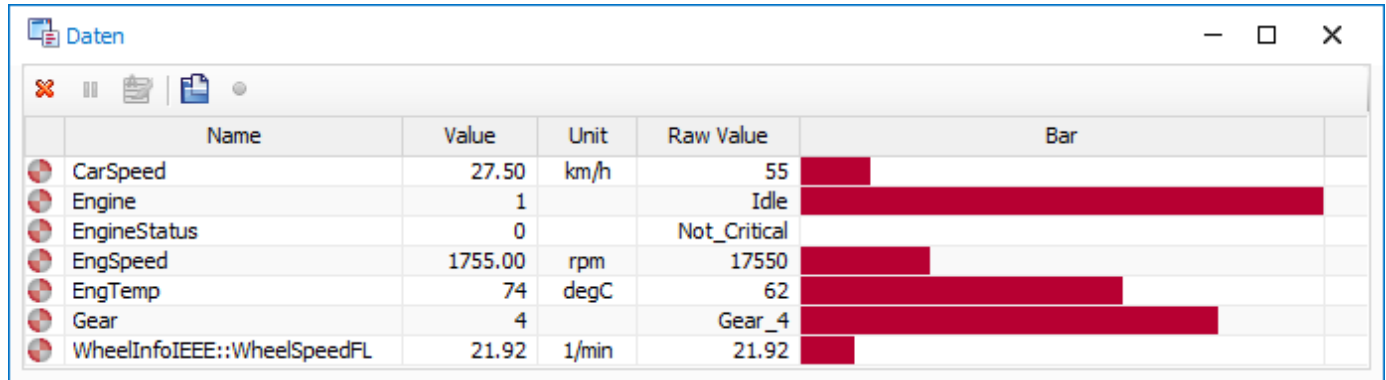
データ Window では、シグナル値、システム変数、診断パラメーターをさまざまな形式で表示できます。

#### > 値の表示

データは生値またはシンボリック値で表示できます。ほかにも指数表現や、最小値／最大値を表示することもできます。

#### > データの記録

測定中にシグナルを記録して MDF バイナリー形式で保存します。



| Name                        | Value   | Unit  | Raw Value    | Bar |
|-----------------------------|---------|-------|--------------|-----|
| CarSpeed                    | 27.50   | km/h  | 55           |     |
| Engine                      | 1       |       | Idle         |     |
| EngineStatus                | 0       |       | Not_Critical |     |
| EngSpeed                    | 1755.00 | rpm   | 17550        |     |
| EngTemp                     | 74      | degC  | 62           |     |
| Gear                        | 4       |       | Gear_4       |     |
| WheelInfoIEEE::WheelSpeedFL | 21.92   | 1/min | 21.92        |     |

図 6: さまざまな表示タイプで取得値を表示するデータ Window

### 3.1.6 統計 Window

統計 Window には、測定中のバスアクティビティに関する統計情報が表示されます (CAN、LIN、FlexRay)。ノードおよびフレーム単位のバス負荷、バーストカウンター／期間、フレームおよびエラーのカウンター／レート、コントローラーの状態などが表示されます。

#### > 個々のチャンネルの統計データを表示

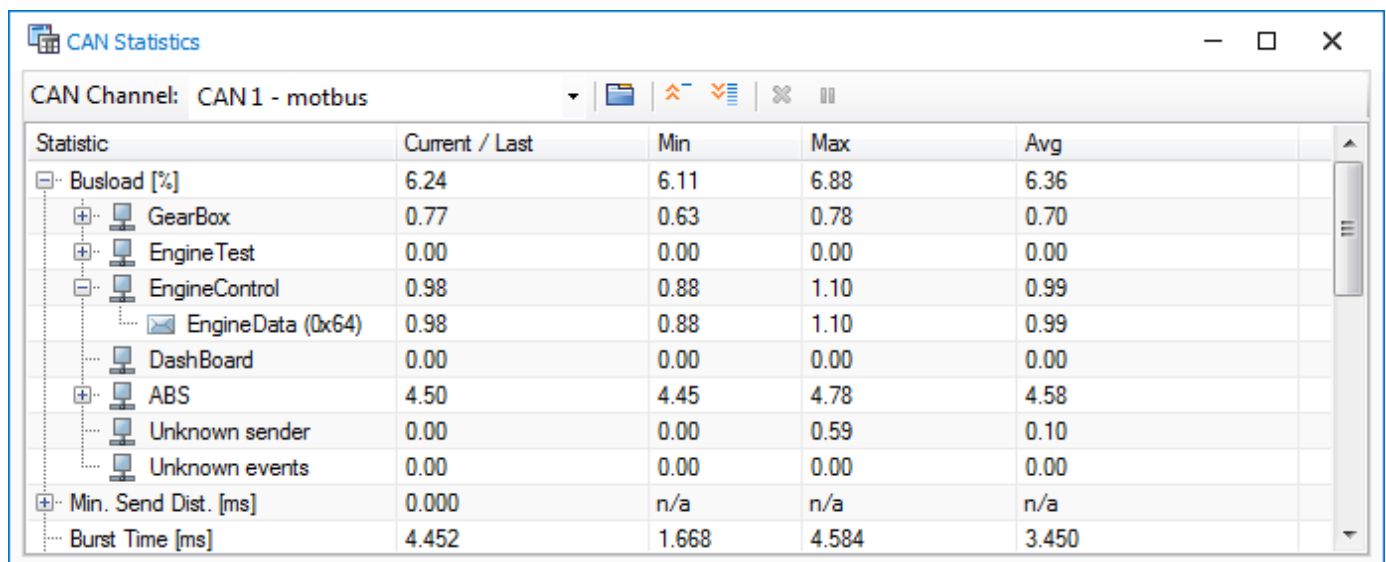
特定チャンネルの統計データのみの表示、あるいは使用可能なすべてのチャンネルの表示を選択できます。

#### > 更新間隔の設定

表示の更新間隔を変更します。

#### > 統計表示の一時停止

測定中に統計データの表示を一時停止できます。



| Statistic            | Current / Last | Min   | Max   | Avg   |
|----------------------|----------------|-------|-------|-------|
| Busload [%]          | 6.24           | 6.11  | 6.88  | 6.36  |
| GearBox              | 0.77           | 0.63  | 0.78  | 0.70  |
| EngineTest           | 0.00           | 0.00  | 0.00  | 0.00  |
| EngineControl        | 0.98           | 0.88  | 1.10  | 0.99  |
| EngineData (0x64)    | 0.98           | 0.88  | 1.10  | 0.99  |
| DashBoard            | 0.00           | 0.00  | 0.00  | 0.00  |
| ABS                  | 4.50           | 4.45  | 4.78  | 4.58  |
| Unknown sender       | 0.00           | 0.00  | 0.59  | 0.10  |
| Unknown events       | 0.00           | 0.00  | 0.00  | 0.00  |
| Min. Send Dist. [ms] | 0.000          | n/a   | n/a   | n/a   |
| Burst Time [ms]      | 4.452          | 1.668 | 4.584 | 3.450 |

図 7: 1 チャンネル (CAN 1) の統計データを表示している CAN 統計 Window

自動的に定義された統計用システム変数を使用して、個々の CAN/LIN/FlexRay 統計データをグラフィック Window などの解析 Window やプログラムノードで評価できます。これらのシステム変数は、各ネットワークチャンネルごとに使用でき、統計 Window とは独立して更新されます。

### 3.1.7 出力 Window

出力 Window には、システムメッセージと CAPL によるユーザー指定の出力が表示されます。

> 出力の設定

出力 Window には各種ビューがあり、出力ソースに従ってシステムメッセージをフィルターします。

> 出力の記録

出力 Window の内容は、ファイルに保存したり、テキストとしてクリップボードにコピーしてそこから他の Windows アプリケーションにコピーしたりできます。

> ステータスの表示

出力 Window 内に、警告やエラーメッセージの情報を通知します。

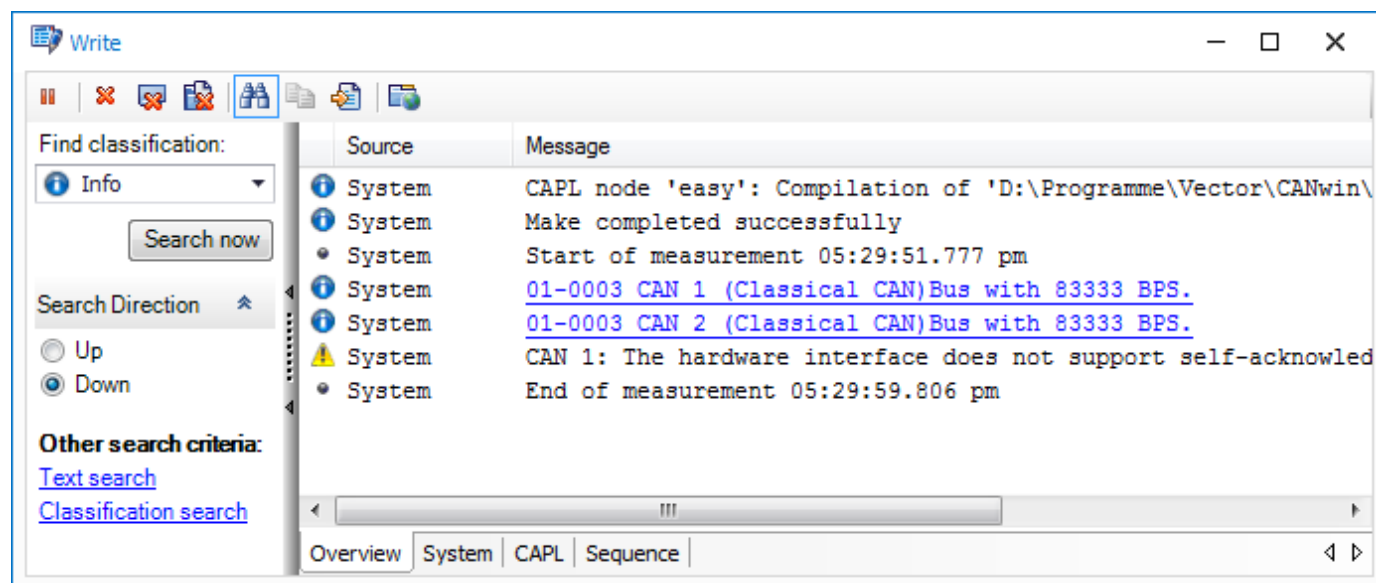


図 8: システムメッセージと CAPL からの出力を表示する出力 Window

### 3.1.8 マップ Window

マップ Window を使用すると、CANalyzer に GNSS 情報とマップを組み込むことができます。マップ Window は CANalyzer の基本機能セットの一部です。オプション Car2x を使用することにより、Car2x 情報を追加表示できます。

- > **データの表示**

GNSS データとマップを解析 Window に表示できます。

- > **オフライン解析**

記録した GPS データを、後からリプレイして解析できます。

- > **Window の同期**

GNSS データは他の解析 Window と同期できます。

- > **地図の表示**

マップ Window では、自車の位置と対象ルートがマップ上に表示されます。これにより、記録された測定データを解析する際に、地理的な条件を考慮することができます。

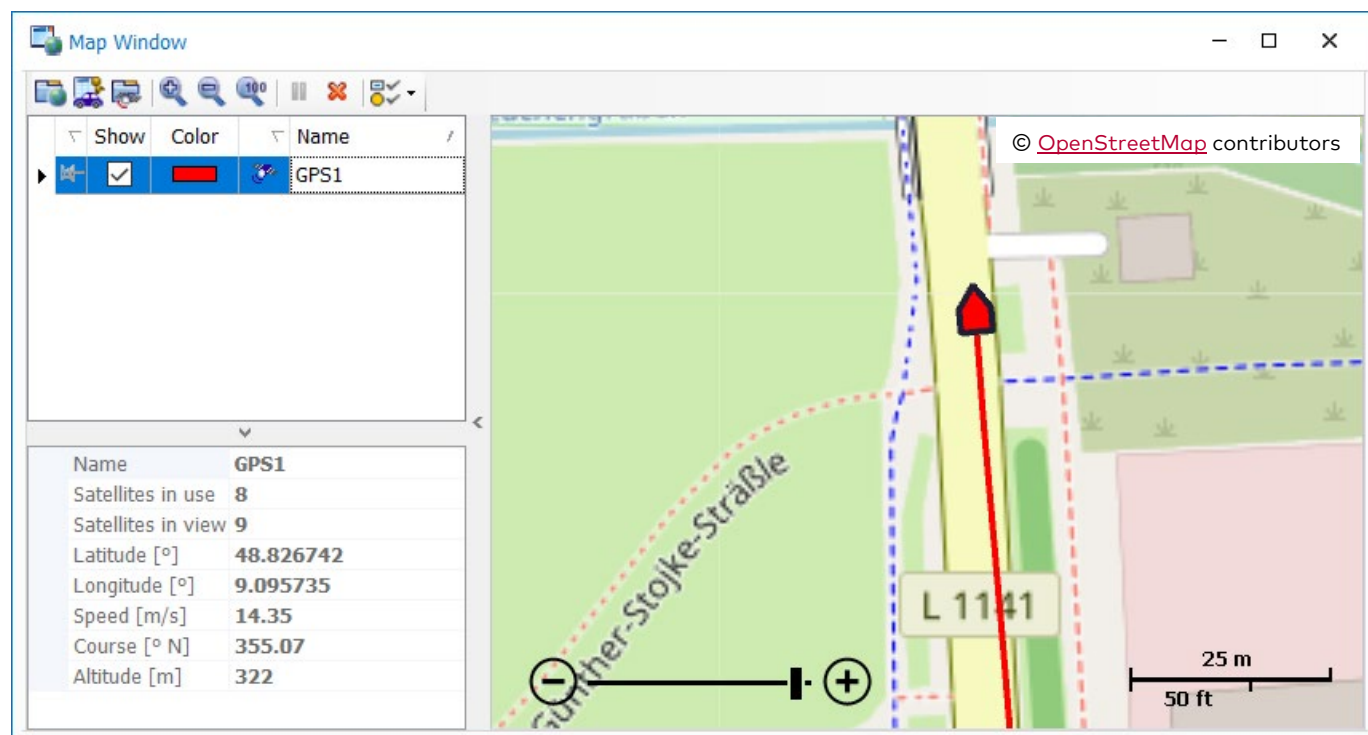


図 9: マップ Window

### 3.1.9 インタラクティブジェネレーター

インタラクティブジェネレーターを使用すると、メッセージを送信するだけでなく、それに対応するシグナル値を設定できます。これにより、バスに対する刺激を容易に行うことができます。

#### > メッセージの定義

メッセージは送信リストを用いて手動で、またはデータベースを使用して定義できます。メッセージの特性はカスタマイズ可能です。

#### > メッセージの送信

特定の画面ボタンやあらかじめ定義されたキーを押したときに、送信リストで設定したメッセージを一定期間ごとに送信することができます。

#### > シグナル値の変更

インタラクティブジェネレーターでは、個々のメッセージの生データをシグナルリストで変更できます。メッセージ内に設定されたシグナルについては、**統合されたシグナルジェネレーター**でシグナル波形(シグナルカーブ)を定義することができます。生データやシグナル値に関連するメッセージで送信して、簡単に ECU の反応を確認したりすることができます。

#### > レイヤー7 プロトコル

インタラクティブジェネレーターは、バスシステムに応じて、単純なレイヤー7 プロトコル(J1939、GMLAN など)と多重化メッセージの伝送をサポートします。

The screenshot shows the CAN IG window with a table of messages and a detailed view of the selected 'WheelInfo' message.

| Row | Send   | Trigger          | Name          | ID  | Channel | Type     | DLC |
|-----|--------|------------------|---------------|-----|---------|----------|-----|
| 1   | [Send] | Manual           | ABSdata       | C9  | CAN 1   | CAN Data | 3   |
| 2   | [Send] | Manual, on ke... | EngineData    | 64  | CAN 1   | CAN Data | 4   |
| 3   | [Send] | Manual, on ke... | EngSpeedContr | 12C | CAN 1   | CAN Data | 1   |
| 4   | [Send] | Periodic: 50 ms  | GearBoxInfo   | 3FC | CAN 1   | CAN Data | 1   |
| 5   | [Send] | Periodic: 100 ms | WheelInfo     | C8  | CAN 1   | CAN Data | 8   |
| 6   | [Send] | Periodic: 200 ms | WheelInfoIEEE | C7  | CAN 1   | CAN Data | 8   |
| 7   | [Send] | Periodic: 100 ms | Raw Frame     | 1   | CAN 2   | CAN Data | 8   |

The 'Basic Properties' panel for the selected 'WheelInfo' message shows:

- Channel: CAN 1
- ID: C8
- IDE: ☐
- Name: WheelInfo
- Trigger: Periodi...
- Type: CAN D...
- DLC: 8
- Protocol: -

The 'Signals' panel shows the 'Raw Data' view for the selected message:

| Name         | Generator Control    | Generator Type | Raw Value | Raw Step | Phys Value | Phys Step | Unit  | Start... | Length |
|--------------|----------------------|----------------|-----------|----------|------------|-----------|-------|----------|--------|
| WheelSpeedRR | [Send] [Stop] [Edit] | User Defined   | 0         | CCD      | 0.00       | 65.00     | 1/min | 48       | 16     |
| WheelSpeedRL | [Send] [Stop] [Edit] | Toggle         | 0         | CCD      | 0.00       | 65.00     | 1/min | 32       | 16     |
| WheelSpeedFR | [Send] [Stop] [Edit] | None           | 0         | CCD      | 0.00       | 65.00     | 1/min | 16       | 16     |

図 10: 設定されたメッセージとそのシグナルを表示するインタラクティブジェネレーター

### 3.1.10 トリガーとフィルター

トリガーとフィルターは、特定のバスイベントに対して反応させることができます。これらを使用すると、表示したり記録したりするデータの量を減らすことができます。トリガー条件の例として、エラー状態、メッセージ、シグナル、シグナルの変化（エッジ）があります。複雑なシステム状態をトリガーするには、グループを形成して、それらを論理演算子を使用して結合します。

#### > 測定設定でのフィルター

測定設定では、さまざまなフィルターを使用して、特定の解析 Window に送られるデータや、ブロックされるべきデータを定義できます。すべてのフィルターはブロックフィルターとパスフィルターとして使用できます。

#### > 測定設定でのトリガー

測定設定では、さまざまなトリガー条件を使用して、ログファイルへのデータ記録を行うことができます。

#### > トレース Window でのフィルター

トレース Window では、各種フィルターを使用して、測定中および測定後の解析データの量を減らすことができます。たとえば、個々のシグナルおよびシグナル値をフィルターするためにフィルターを定義したり、さまざまなカラムフィルターを設定したりできます。

#### > ハードウェアでのフィルター

CAN コントローラーはアクセプタンスフィルターにより、どの受信メッセージが CANalyzer に送られるかを制御します。

### 3.1.11 ロギング／リプレイ

CANalyzer でデータを記録して、後で解析用途にリプレイできます。

#### > リプレイ

リプレイブロックを使用して、ログファイルに記録した測定シーケンスをリプレイできます。ログファイルに記録されたメッセージはデータフローに出力されます。

#### > ログ記録

ログブロックは、バストラフィックを BLF および ASCII 形式で記録できます。記録したデータは、オフラインモードでリプレイしたり、リプレイブロックを使用してリプレイできます。

### 3.2 データの解釈

CANalyzer には、ISO/DIS 15765-2 トランスポートプロトコル用のオブザーバーが含まれています。このオブザーバーは、CAN バス上でトランスポートプロトコルメッセージを解釈し、その結果をテキスト形式でトレース Window に表示します。さらに、KWP2000、UDS または OEM 固有の CANdelaStudio データベースファイル(CDD)、ODX/PDX および MDX ファイルに基づいて診断解釈を行うこともできます。CANalyzer で定義したベーシックダイアグノスティックの記述は、診断解釈にも使用できます。トレース Window での解釈に加えて、グラフィック Window およびデータ Window に個々の診断パラメータを表示することもできます。

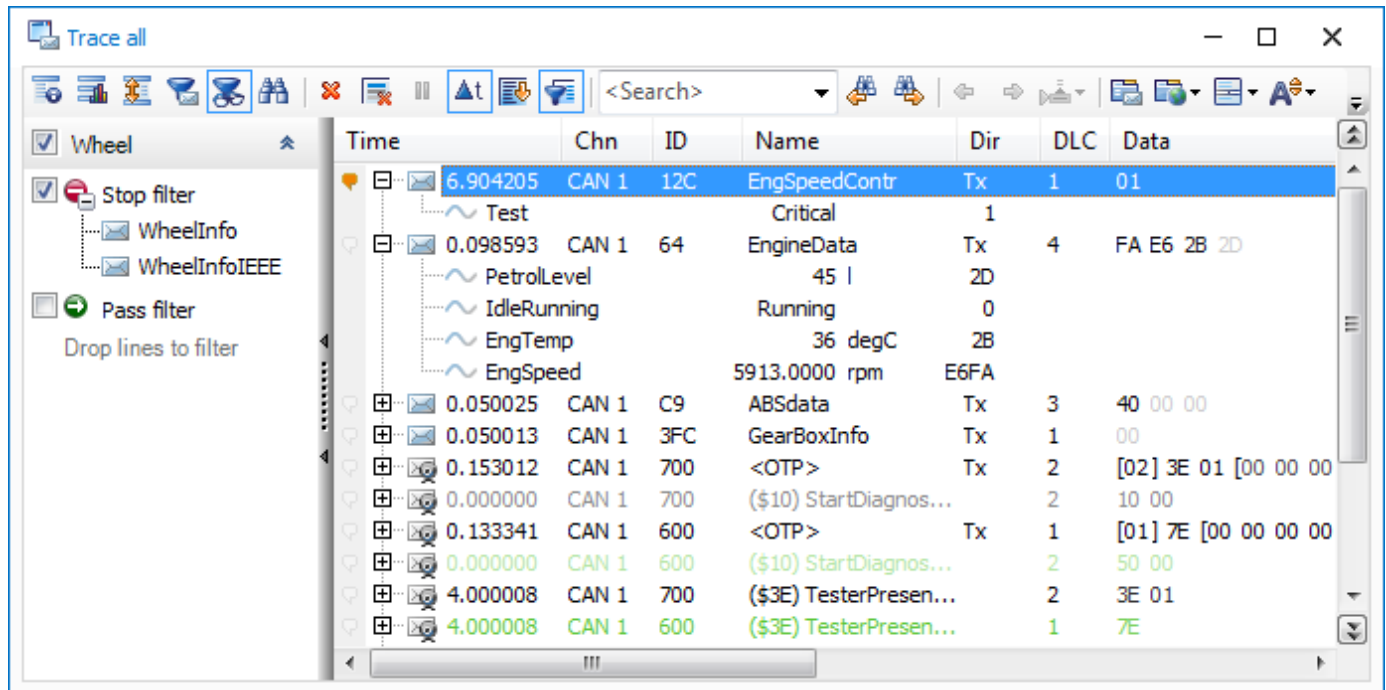


図 11: 解析フィルターと診断データベースファイルによる診断データが表示されたトレース Window

### 3.3 オフライン評価

ログに記録されたメッセージトラフィックは、CANalyzer の全機能を使用できる、オフラインモードで評価することができます。これによって、たとえば机上であっても、テスト車両運転時の広範な記録の解析がしやすくなります。このような解析にはログファイルの複数使用も可能であり、タイムスタンプに基づいて時系列で解析に組み込みます。また、ログファイルを直接トレース Window にインポートすると、すぐに解析することができます。

オフラインモードでは、ベクターの GL1000/GL2000/GL3000/GL4000 ロガーファミリーおよび CANlog のログ形式を直接サポートします。これにより、解析前の変換ステップが軽減されます。

### 3.4 エクスポート機能

ログおよびトレース Window とグラフィック Window のエクスポート機能を使用して、記録されているログファイルやグラフィック Window の内容を、その他のファイル形式(\*.csv など)に変換することができます。ログファイルをエクスポートする際、個々のシグナル情報はメッセージ指向のログとして記録されたデータから抽出されます。

### 3.5 統合デスクトップ

CANalyzer での作業中には、多数の解析 Window を開いたままにしておくことがよくあります。このため、CANalyzer には統合デスクトップ機能が備えられており、ページタブを選択して任意の数の仮想デスクトップ間を切り替えることができます。開いている Window を複数のデスクトップに分散したり、解析情報を作業プロセスやトピック別に分類することができます。

### 3.6 開始値の設定

開始値 Window では、測定開始時にシステム変数に設定される値をあらかじめ割り当てることができます。

開始値のリストをファイルにエクスポートしたり、ファイルから読み込んだりできます。たとえばシミュレーションパラメーターなどを、多様な開始値のセットを使用して簡単に割り当てることができます。

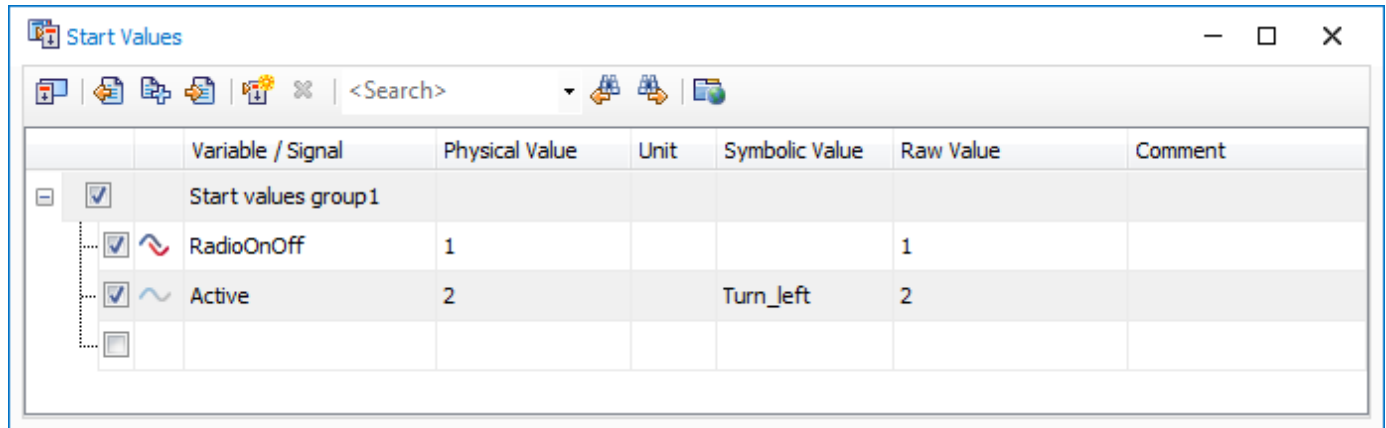


図 12: 開始値 Window

### 3.7 シンボルマッピング

シンボルマッピングダイアログを使用して、システム変数のマッピングを行うことができます。

測定中にソースの値が変わると、それに対応する変数の値も自動的に設定されます。線形変換公式を適用することもできます。

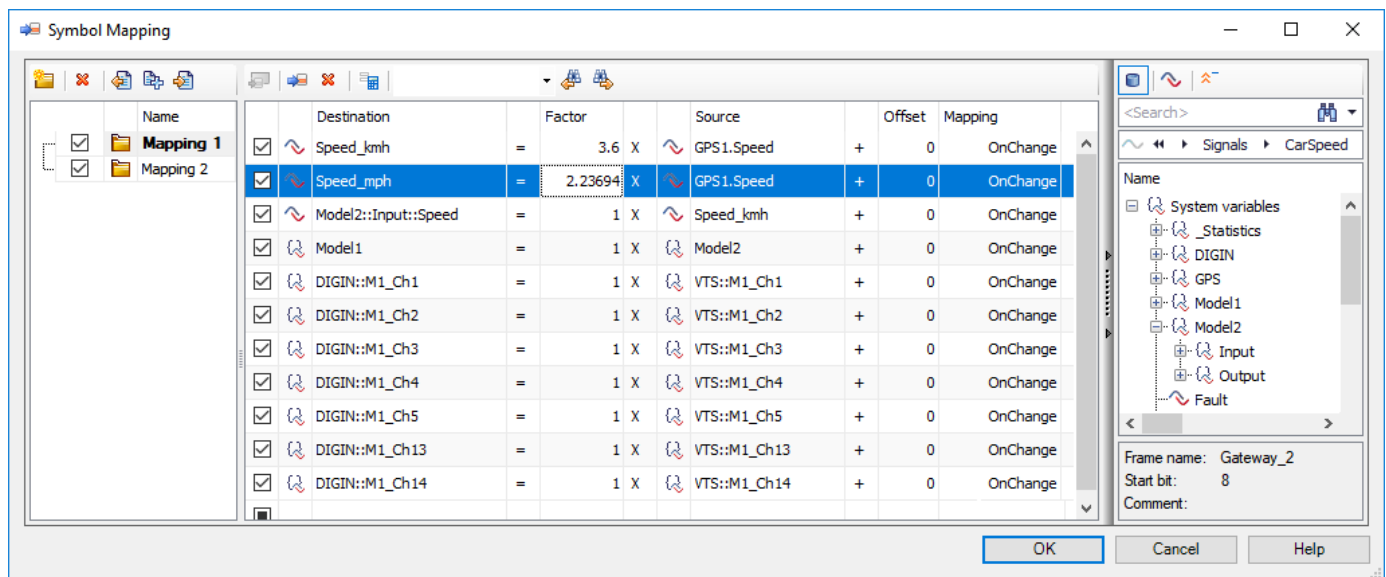


図 13: マッピングダイアログ

## 4 診断

すでに KWP2000 または UDS の標準診断サービスが定義されている場合、それを CANalyzer から送信できます。さらに、診断パラメーターを定義して、Local ID (LID) を割り当てることもできます。ECU の情報は、診断コンソールからすばやく簡単に読み取ることができます。

以下のようなコンセプトと機能を使用できます。

- > KWP2000 および UDS (ISO 14229) 用の診断データベースフォーマットをサポート:
  - > ODX 2.0.1/2.2.0(PDX ファイル)
  - > MDX 2.0/3.0/4.0
  - > CANdelaStudio (CDD)
- > 診断/ISO-TP 設定ダイアログで診断データベースファイルの主な診断通信パラメーター(トランスポートレイヤーおよび診断レイヤー)を変更可能
- > 診断データベースファイルを使用できない場合に、簡単な診断サービスを迅速に定義できるベーシックダイアグノスティックエディター(CAN、LIN、FlexRay、Ethernet、K-Line に対応)
- > 診断コンソール、フォールトメモリーWindow にて、セキュリティDLL を使用可能なインタラクティブ診断テスター
- > セキュリティーソースによって設定されたセキュリティメカニズムを考慮してバリエーションコーディングデータの読み込み、書込み、比較を行う、インタラクティブなバリエーションコーディング Window
- > 周期的または対話形式で診断リクエストを送信し、指定した診断レスポンス内のパラメーターを表示するための診断パラメーターWindow
- > 設定済みの OBD-II テスターと、OBD-II 用の診断コンソールとフォールトメモリーWindow
- > 複数のアドレッシング手法(Normal、Extended、Normal Fixed、Mixed など)とアドレッシングタイプ(Functional/Physical)をサポート
- > トレース Window、データ Window、グラフィック Window で、サービスとパラメーターレベルで診断通信を解析(診断データベースファイルに基づくシンボリック表示)
- > トレース Window にプロトコルエラーを表示
- > パネルを使用した診断パラメーターの表示および診断リクエストによる ECU への刺激入力
- > 自動車業界で利用されている主なネットワークタイプをサポート(CAN、LIN、FlexRay、Ethernet、K-Line)
- > DoIP (Diagnostics over IP)、HSFZ (High Speed Fahrzeugzugang)、DoSoAd (Diagnostics over AUTOSAR Socket Adaptor) のサポート
- > マクロを利用した診断シーケンスの記録とリプレイ

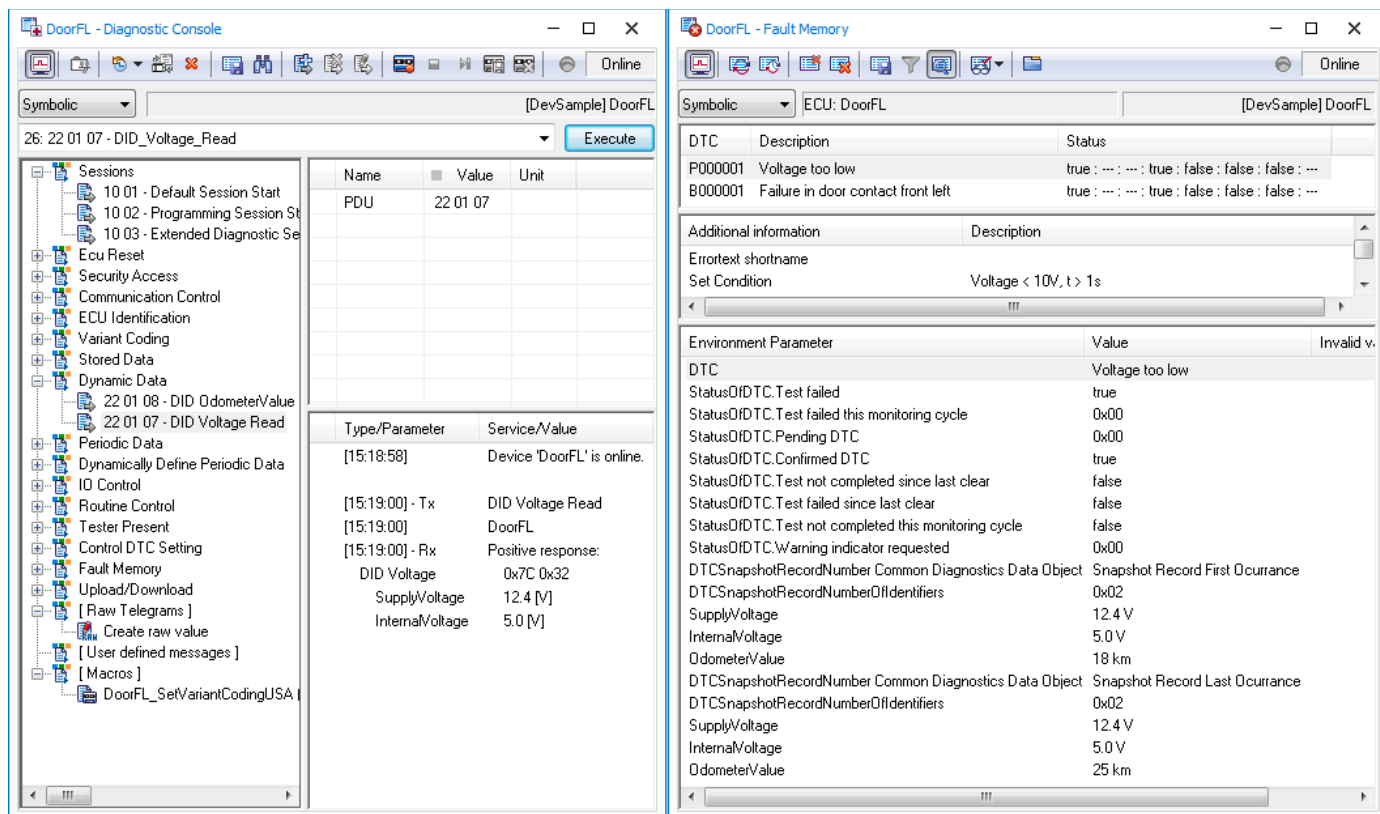


図 14: 診断コンソールおよびフォールトメモリーWindow

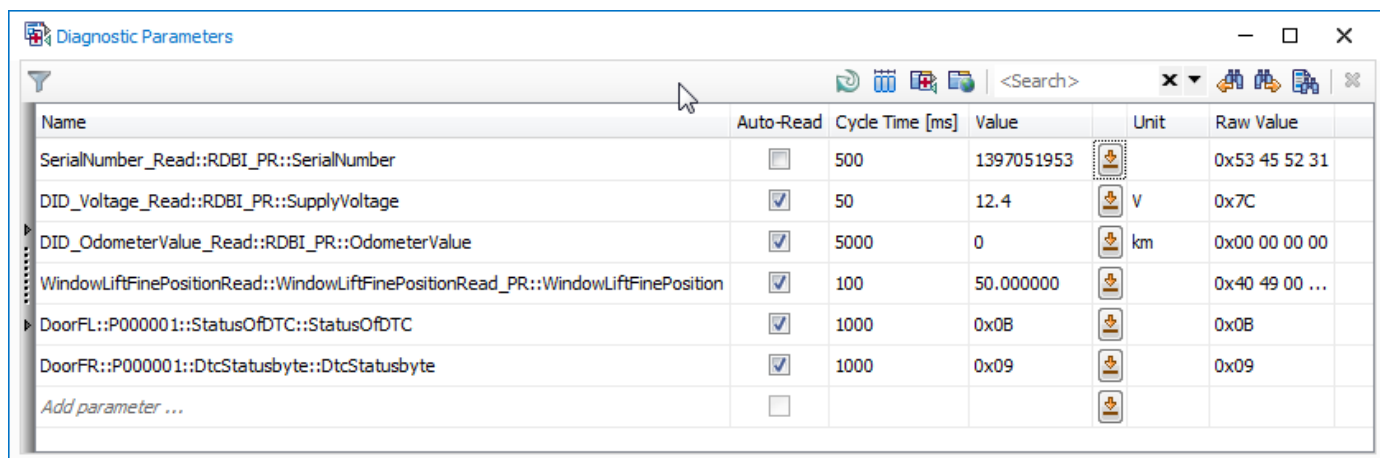


図 15: 診断パラメーターWindow

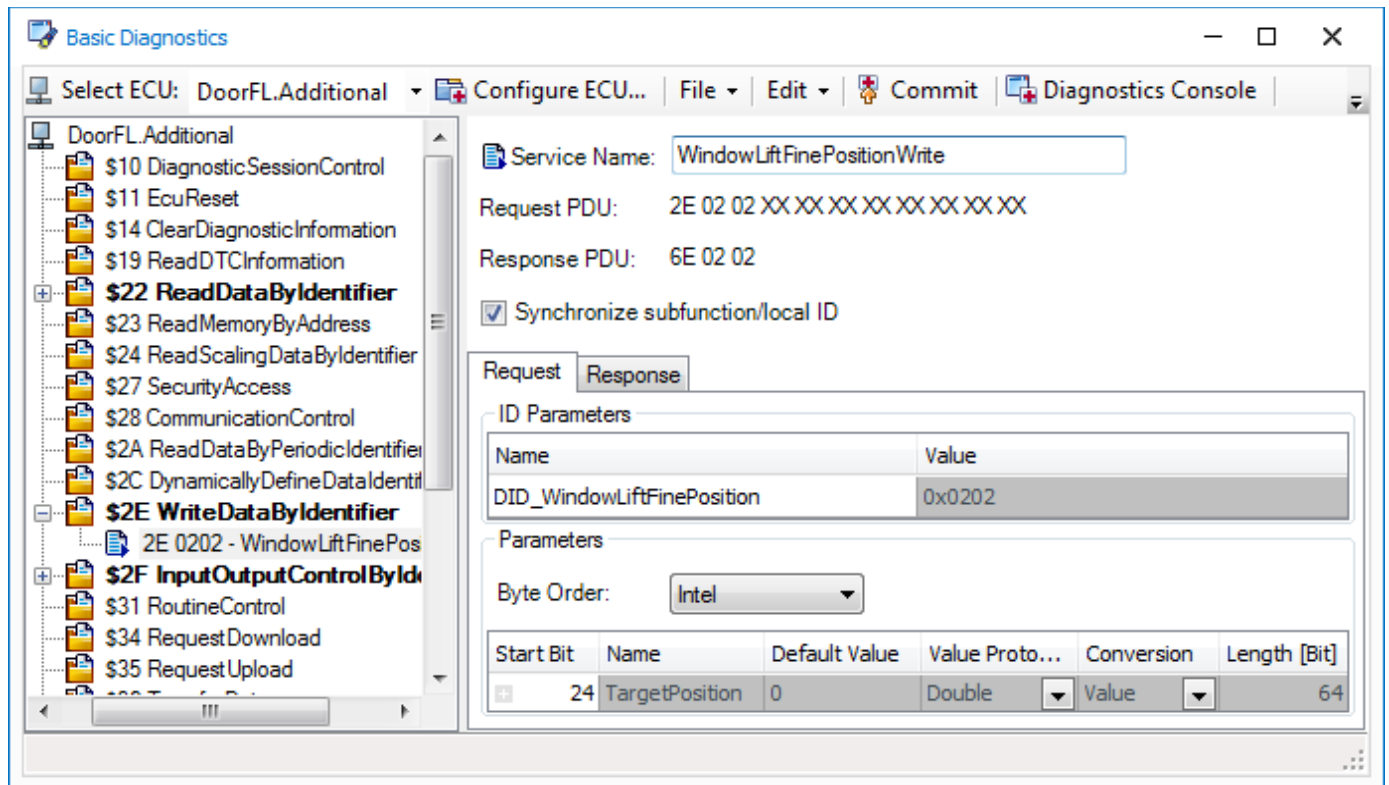


図 16: ベーシックダイアグノスティックエディター

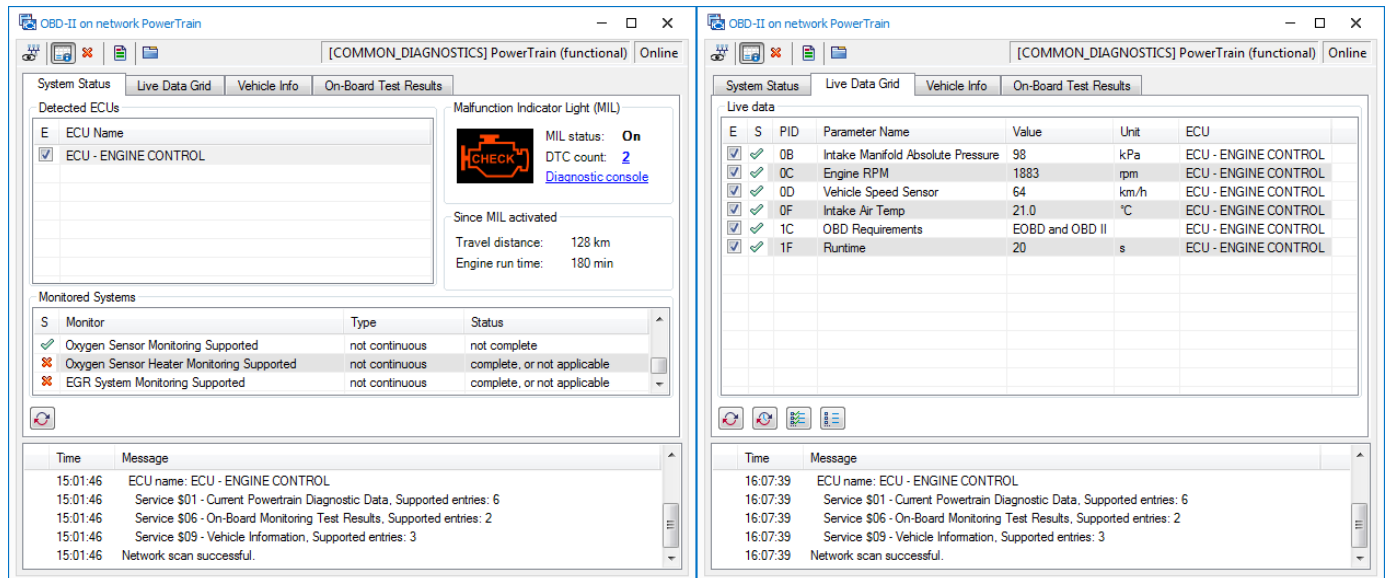


図 17: OBD-II Window

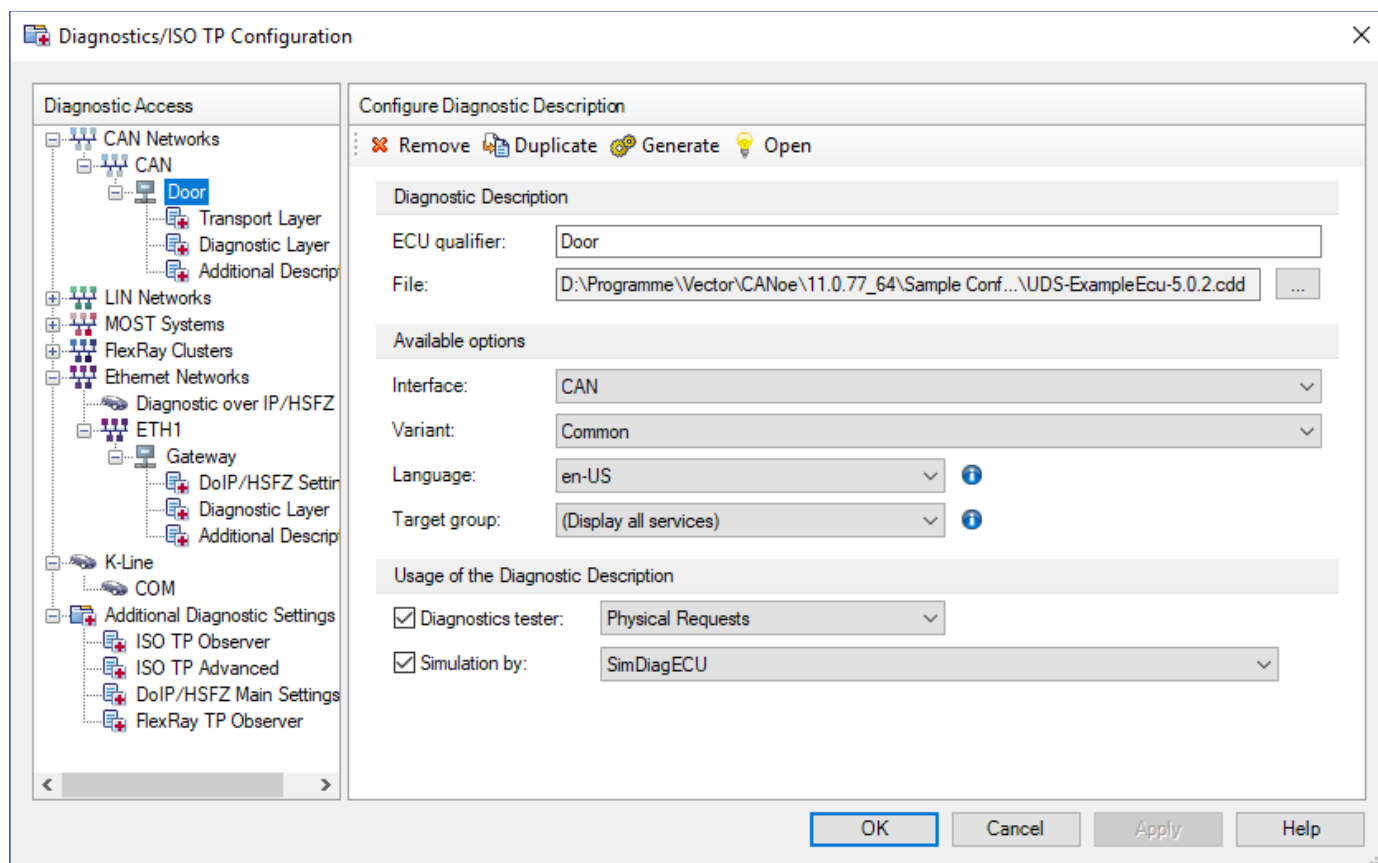


図 18: 診断／ISO-TP 設定ダイアログ

|                              |       |                                         |       |          |          |   |             |
|------------------------------|-------|-----------------------------------------|-------|----------|----------|---|-------------|
| 14.116307                    | 27 01 | Seed Level #1 Request::req              | req   | <tester> | Door     | 2 | 27 01       |
| 14.117771                    | 67 01 | Seed Level #1 Request::pos              | pos   | Door     | <tester> | 4 | 67 01 48 0A |
| Variant: "CommonDiagnostics" |       |                                         |       |          |          |   |             |
| SID-PR                       |       | 0x67                                    | 67    |          |          |   |             |
| Type                         |       | 0x01                                    | 01    |          |          |   |             |
| SecuritySeed                 |       | 0x480A                                  | 48 0A |          |          |   |             |
| 0000:                        |       | 67 01 48 0A                             | g.H.  |          |          |   |             |
| 14.142271                    | 27 02 | Key Level #1 Send::req                  | req   | <tester> | Door     | 4 | 27 02 B7 F5 |
| Variant: "CommonDiagnostics" |       |                                         |       |          |          |   |             |
| SID-RQ                       |       | 0x27                                    | 27    |          |          |   |             |
| Type                         |       | 0x02                                    | 02    |          |          |   |             |
| SecurityKey                  |       | 0xB7F5                                  | B7 F5 |          |          |   |             |
| 0000:                        |       | 27 02 B7 F5                             | '...  |          |          |   |             |
| 14.143759                    | 67 02 | Key Level #1 Send::pos                  | pos   | Door     | <tester> | 2 | 67 02       |
| 25.552319                    | 19 02 | Fault Memory Read (all identified)::req | req   | <tester> | Door     | 3 | 19 02 00    |

図 19: トレース Window での診断通信の表示

## 5 プログラミング

### 5.1 CAPL インターフェイス

CAPL (Communication Access Programming Language) プログラミング言語は、CANalyzer の機能範囲を大幅に拡張します。CAPL には以下のような特長があります。

- > C プログラミング言語をベースとするため習熟が容易
- > CANalyzer がユーザーに代わり、オペレーション内で完全にイベント制御
- > メッセージやシグナルなどの、すべてのデータ情報へのシンボリックアクセスをサポート。シグナル値を物理形式で使用可能
- > 特殊な関数により言語を拡張し、さまざまな使用シナリオで素早く問題を解決
- > 外部ライブラリーによる柔軟な拡張

#### 5.1.1 C ライクな構文

通常のスカラーデータタイプと配列を利用できます (1、2、4、8 バイト長のタイプおよび 8 バイト長の浮動小数点タイプ)。割り当て、算術演算子、ループフロー制御は C 構文に準拠します。

```
myFunction {
    int counter;

    for ( counter = 0; counter < 8; counter++ ) {
        doSomethingWithCounter ( counter );
    }
}
```

#### 5.1.2 イベント指向制御

CAPL は、イベント制御プログラミング言語です。C と比べ、CAPL では特別にあらかじめ定義されたイベントハンドラー (イベントプロシージャー) を使用でき、これらは特定のイベントが発生すると必ず実行されます (タイマー制御の場合には、ハードウェアまたは CANalyzer 内部のタイマーによりトリガーが発生します)。

以下に、これらのイベントハンドラーの例をいくつか紹介します。

| イベントハンドラー              | イベント             |
|------------------------|------------------|
| On timer seconds cycle | タイマー制御           |
| On message ESPStatus   | メッセージの入出力        |
| On signal update       | シグナル値の更新         |
| On sysvar              | システム変数の変更        |
| On diagRequest         | 診断リクエスト          |
| On FLError             | FlexRay バスエラーの検出 |

### 5.1.3 シンボリックアクセス

シグナル値は、メッセージ全体としてのデータ内容に関係なく、通常は物理値としてアクセスされます。これはデータベースで設定されています。

> メッセージベースでのアクセス:

```
message EnergyMgmt msg;

// Most significant bytes Motorola, with 12bit only the lower 4 bits are used
msg.byte(0) = (msg.byte(0) & 0xF0) | (byte)((14.1 - 8) / (18 - 8) * 4096 / 256) & 0xF;

// Least significant byte
msg.byte(1) = (byte)((14.1 - 8) / (18 - 8) * 4096) & 0xFF;

output(msg);
```

### 5.1.4 アプリケーション固有の言語拡張

CANalyzer は、日々の開発業務における問題に特化したさまざまな機能を備えています。

> 解析

CAPL は、測定結果のオンライン解析にもオフライン解析にも使用できます。1 つの簡単なタスクで、特定のイベントの出現をカウントしたり、特定のシグナルの内容を使用して計算を実行することができます。

```
On message Brake {
    long TempCounter = 0;
    $BRECounter++;

    // Weighing the average
    TempCounter = $BRECounter;

    if ($BRECounter > 1000)
        TempCounter = 1000;

    @AveragePressure = @AveragePressure * TempCounter + $Brake::Pressure;

    @AveragePressure = @AveragePressure / (TempCounter + 1);

    output ( this );
}
```

> シミュレーション

CAPL を使用して、ECU を刺激するメッセージの生成も可能です。これによって開発者のルーチンワークが軽減されます。バスのシグナルとメッセージはデータベース(DBC、LDF、FIBEX ファイルなど)に定義されています。これらのファイルは通常、一箇所で管理、保守、更新されます。

### 5.1.5 CAPL ブラウザー

CAPL ブラウザーは、単なる CAPL プログラムのエディターではなく、高度な開発環境の構築をサポートします。

- > コード記述中に、コードのオートコンプリートと構文チェックを実行
- > 構文の強調表示を設定可能
- > 構文を区別するタブ
- > イベントハンドラーおよび関数リファレンスをツリービューで折りたたみ見やすく表示
- > 個々または複数のファイルでの検索および置換
- > オンラインヘルプでの関数検索
- > エラー時に、コンパイラーからエラー対象のソーステキストへ移動可能
- > 関数リストから検索を行い、ソーステキストとして入力可能

CANalyzer データベースのオブジェクトは CAPL ブラウザーでも使用可能で、ツリービューにも表示されます。以下のデータベース内容に、**シンボルエクスプローラー**でアクセスできます。

- > ノード、メッセージ、シグナルなどのネットワークシンボル
- > CANalyzer 全体で使用するシステム変数
- > リクエスト、レスポンス、フォールトメモリーなどのすべての診断シンボル

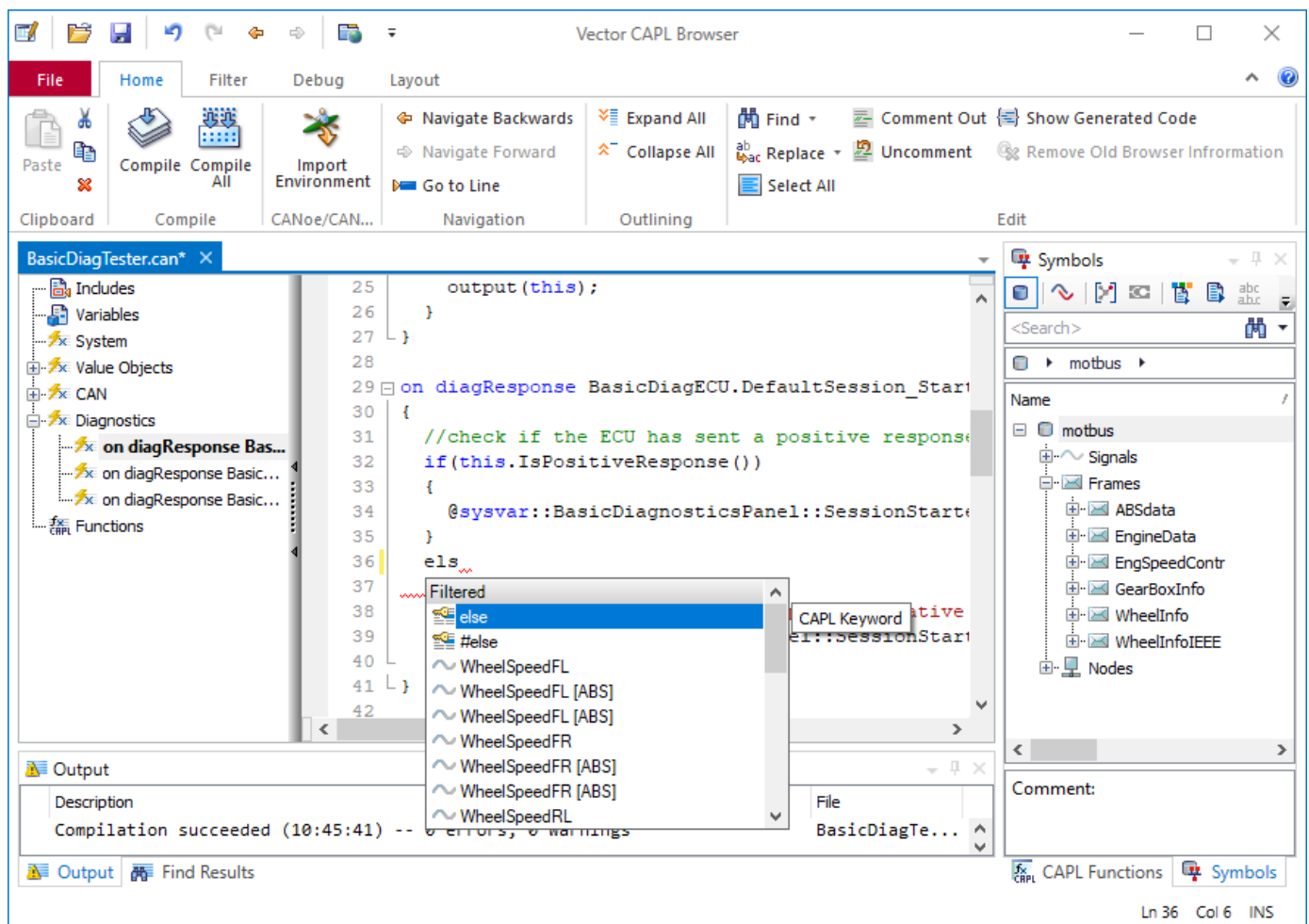


図 20: CAPL ブラウザーで CAPL プログラムを開いた際には、イベントプロシージャーとデータベースのネットワークシンボルが表示される

## 5.2 ビジュアルシーケンサー

プログラミングを行わなくても、動作シーケンスをグラフィカルに構成できます。これらのシーケンス内で、変数とシグナルを設定することもできます。CAN/LIN フレームや診断コマンドも送信できます。また、特定のイベントの待機、値のチェック、制御構造 (repeat...until) を使用した反復定義も可能です。これらのシーケンスは、異機種が混在するシステムのテストや ECU の刺激入力を容易に行うために最適です。

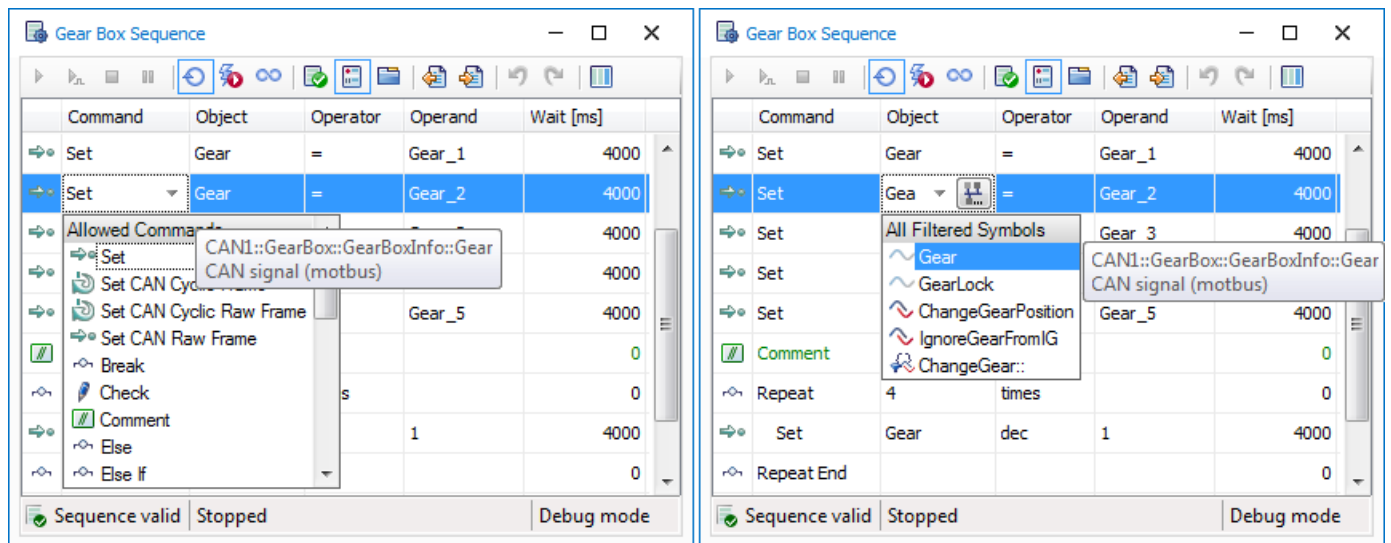


図 21: 簡単なテストおよび刺激入力シーケンスを作成する CANalyzer のビジュアルシーケンサー。

「オートコンプリート」のサポート、データベース情報の詳細表示により、コマンドやオブジェクトのデータベースの選択が容易

## 6 パネル

パネルはグラフィカルなエレメントを使用して、シグナルや変数の値を変更したり、それらをスライダーやポインターなどのコントロールを用いて表示することができます。たとえば、パネルを使用して、CAPL プログラムから解析データを表示したり、CAPL プログラムを制御するための値をプログラムに転送したりできます。

パネルデザイナーでは、パネルを簡単に作成することができます。たとえば、シンボルとコントロールをドラッグ & ドロップ操作で簡単にリンクできます。プロパティ Window を用いて、個々のパネルとコントロールを設定できます。また、最適なレイアウトが行えるように、多様なレイアウト機能が揃っています。

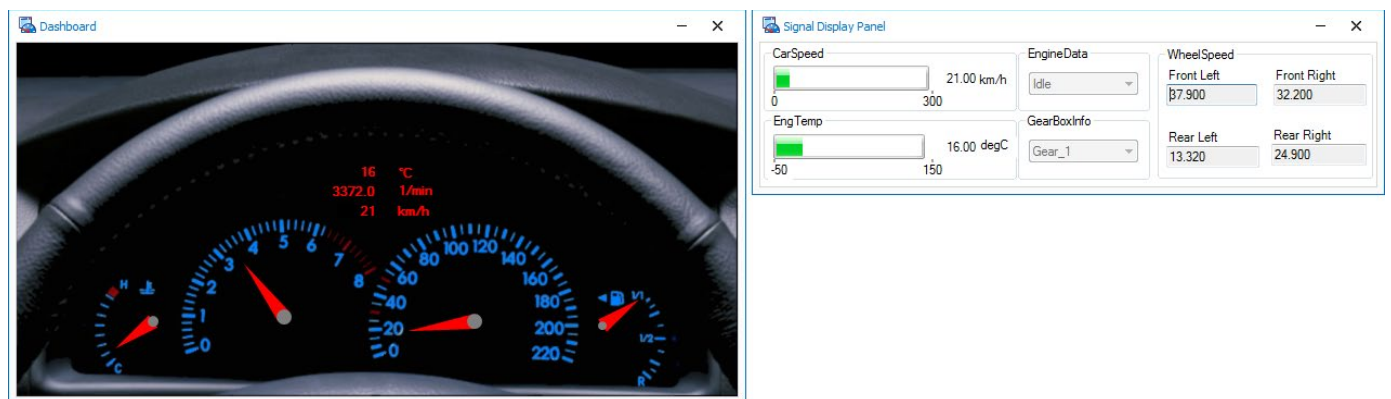


図 22: シグナルと変数の値を表示するユーザー定義のパネル

## 7 ソフトウェアインターフェイス

また、統合された COM (Component Object Model) サーバーにより、外部アプリケーションによって測定シーケンスを制御することができるほか、たとえば測定データの解析や観察されたバストラフィックの詳細評価などのために、標準のソフトウェアとのデータ交換を簡単に行うことができます。よく使用されるプログラミング/スクリプト言語は、Visual Basic または Visual Basic for Applications です。C++/C# もよく使用されます。CANalyzer が COM インターフェイス経由で提供する機能には、以下のようなものがあります。

- > シミュレーションの制御、測定の開始および停止
- > 既存コンフィギュレーションの読み込み、新規コンフィギュレーションの生成、送信ブランチへのデータベースおよびブロックの追加
- > シグナルおよびシステム変数へのアクセス、CAPL 関数へのアクセス、CAPL ノードのコンパイル

測定を開始する際の Visual Basic Script の例:

```
set app = createobject( "canalyzer.application")
set measurement = app.measurement
measurement.start
set app = nothing
```

コンフィギュレーションファイルを開く際の Visual Basic Script の例:

```
set app = createobject( "canalyzer.application")
app.open "D:\PathToMyConfig\myconfig.cfg"
set app = nothing
```

## 8 ネットワークインターフェイス

CANalyzer は、ベクターで提供しているすべてのネットワークインターフェイスをサポートします。さまざまなコンピューターインターフェイス (PCMCIA、USB 2.0、PCI、PCI-Express、PXI) やバストランシーパーに対応しており、あらゆるユースケースにおいて最適なバスアクセスが可能です。

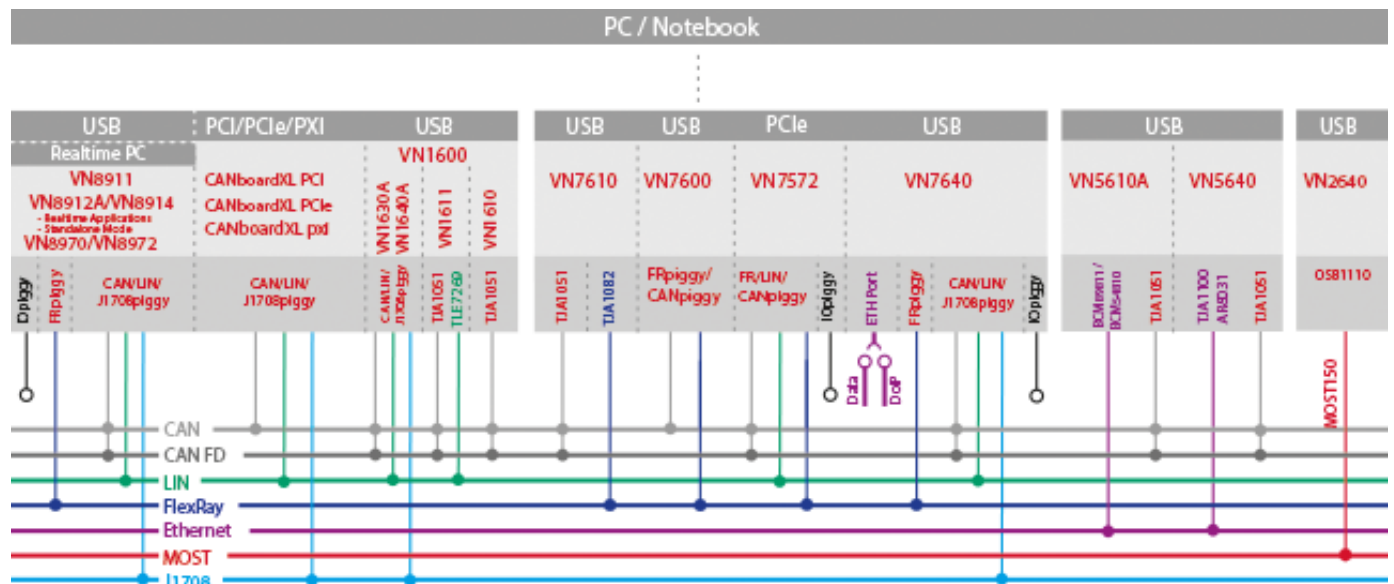


図 23: ベクター製ハードウェアの概要

## 9 オプション Scope

オプション Scope は、極めて高性能な USB 接続オシロスコープハードウェアを使用した、CANalyzer 用の統合オシロスコープソリューションです。この CANalyzer オプションは、プログラム内では設定、バスレベル、プロトコルデコードの各ビューを含む、解析 Window として表示されます。対応するハードウェアは、CAN/CAN FD/FlexRay x 2 または LIN x 4 用に最大 4 つの入力チャンネルを装備し、ベクターのネットワークインターフェイスの (VN1630A/40A、VN8900、VT システムなど) sync ラインによってトリガーされます。オプション Scope は、CANalyzer の全グレードで使用できます。

### 9.1 適用分野

この USB 接続オシロスコープと CANalyzer の強力な連携により、さまざまな方法でプロトコルエラーを解析できるようになります。バス固有のトリガー条件と CANalyzer 時間同期を使用すれば、従来のオシロスコープを使用するよりも格段に速くプロトコルエラーの原因を発見することができます。

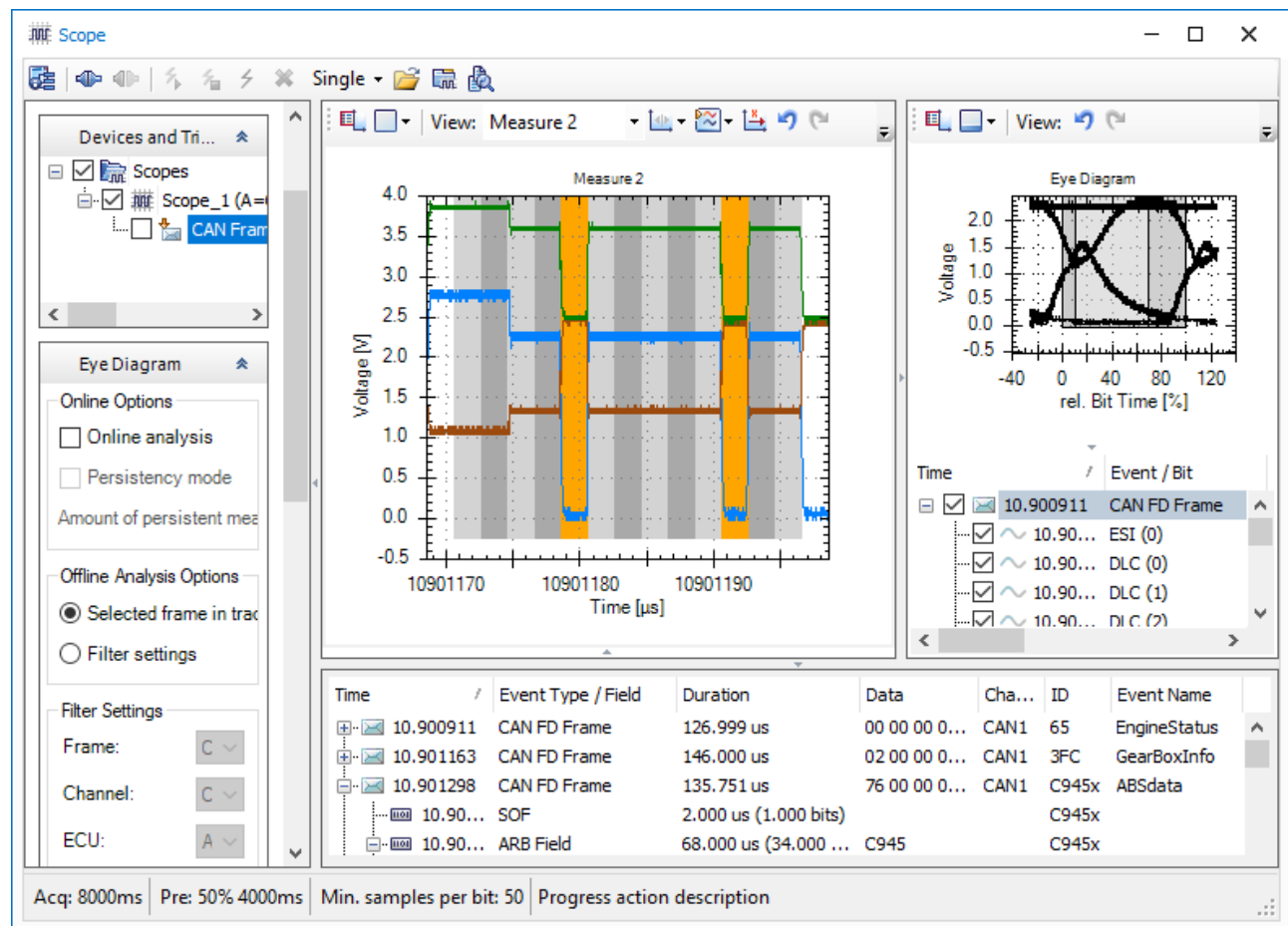


図 24: スコープ Window を使用した物理レベル、論理レベルでの CAN FD フレーム/エラーの詳細解析

### 9.2 主な特長

- > USB 接続オシロスコープハードウェアを使用した、コンパクトで携帯可能なオシロスコープソリューション
- > Vector Scope Bus Probe による簡単なバス接続
- > 特にバス解析向けに開発されたオシロスコープ機能。たとえば、
  - > バスエラーおよび特定イベント向けに事前定義可能なトリガー条件
  - > CANalyzer タイムベースと同期したバスおよび I/O シグナルの記録
  - > プロトコルエラー (CAN エラーフレームなど) をデコード

### 9.3 サポートするプロトコル

- > バスシステム: CAN、CAN FD、FlexRay、LIN

## 9.4 対応するオシロスコープハードウェア

- > Vector PicoScope 5444B-034:
  - > 帯域幅 200MHz、サンプリングレート 500MS/秒の、高精度の USB 接続オシロスコープ
  - > 4 系統のバスシグナル入力チャンネル(CAN/CAN FD/FlexRay x 2 または LIN x 4)
- > Vector PicoScope 5242B-034:
  - > 帯域幅 60MHz、サンプリングレート 500MS/秒の、高精度の USB 接続オシロスコープ
  - > 2 系統のバスシグナル入力チャンネル(CAN/CAN FD/FlexRay x 1 または LIN x 2)
- > Vector Scope Bus Probe を通じて D-Sub コネクタとバス接続
- > Vector Scope Y-Trigger Cable により、ベクターネットワークインターフェイスの sync ラインによる内部トリガーに加えて外部トリガーも実行可能

## 9.5 オシロスコープソフトウェア

CANalyzer では、オプション Scope は設定／測定、バスレベル、プロトコルデコードの各ビューを含む新規の解析 Window として表示されます。

### 9.5.1 設定機能

- > 複数のバスシグナルをオシロスコープの入力チャンネルに接続。バスシステムごとにオシロスコープハードウェア用のデフォルト設定が作成される
- > バスのボーレートに関わらず、ビットあたりの最小サンプル数を基にサンプリングレートを簡単に設定
- > バスのボーレートに応じて波形データ取得時間を自動調整
- > Pre/Post Trigger 時間は 10%から 90%まで調整可能
- > 同一オシロスコープデバイスの複数接続をサポート

### 9.5.2 トリガー機能

- > トリガーモードは**シングル／リピート**を選択可能
- > ツールバーを使用した手動、または CAPL 関数によるトリガー
- > 単純なトリガー条件:
  - > フレームに基づくトリガー(ID または ID 範囲)
  - > プロトコルエラーに基づくトリガー(CAN エラーフレームなど)
  - > すべての条件を OR 条件で結合可能
  - > 外部シグナルのエッジ、またはパルスによるトリガー(IO トリガー)
- > 高度なトリガー条件:
  - > メッセージ内シグナルまたはシステム変数の変化に基づくトリガー(例: 入出力シグナルの変化)
  - > 複雑なトリガー条件を AND/OR 条件で定義可能
  - > CAPL の COM インターフェイスを介したトリガー条件の設定

### 9.5.3 解析機能

- > プロトコルエラーが発生したフレームでも、バスレベルをビットレベルで詳細にデコード
- > 各サンプリングポイントでタイムスタンプと電圧値を表示
- > トレースビュー（データリンクレイヤーの論理値）とダイアグラム（波形）の双方向同期。ダイアグラムにはトレースビューのシグナルのエンコーディングを表示
- > トレース Window、グラフィック Window、ステートトラッカーなど、その他の CANalyzer 解析 Window との時間同期

### 9.5.4 オフライン機能

オプション Scope のライセンスを保有しないユーザーでも、測定済みデータの表示や解析を行えるようにするため、オフラインモードでスコープ Window の機能が利用可能です。

- > スコープ測定データの概観と管理機能
- > 新しい測定データの自動表示
- > ユーザーインターフェイスから設定可能な、単一のビットマスクによるアイダイアグラム解析
- > スコープ測定データの完全なエクスポートおよびインポート（バイナリー形式\*.csf）
- > スコープ測定データの ASCII (\*.csv) または MATLAB (\*.mat) 形式でのエクスポート
- > アイダイアグラムのエクスポート／インポート(\*.ceye)
- > テストレポートに利用できるスクリーンショットの作成とビットマップのエクスポート
- > グローバルマーカを使用して、測定の重要ポイントをマーク

## 10 トレーニング

ベクターでは、CANalyzer に関するトレーニングを開催しています。また、お客様ご指定の場所でのオンサイトトレーニングも実施しています。各コースの詳細やスケジュールにつきましては、ベクターの下記 Web サイトをご覧ください。

[https://vector-academy.com/vj\\_training\\_ip.html](https://vector-academy.com/vj_training_ip.html)

