INTREPID CONTROL SYSTEMS, INC.

Vehicle Spy

Help 日本語版

序章～チュートリアル編

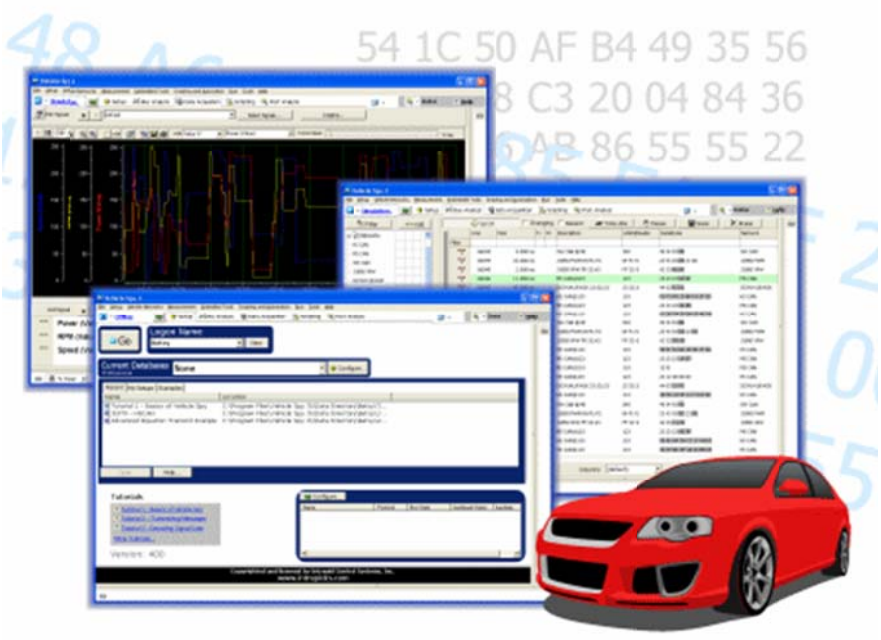
目次

1. Welcome! 4
2. イントロダクション 5
2.1. 概観 5
2.2. 要点 6
2.2.1. バストラフィック観測 6
2.2.2. シグナル観測 7
2.2.3. フライトレコーダ 8
2.2.4. ノード・車両シミュレーション 9
2.2.5. テスト自動化 10
2.2.6. まとめ 11
2.3. 世界のユーザ様一覧 11
2.4. 仕様 12
2.5. 要件 15
2.6. ウェブサイト 16
3. チュートリアル 17
3.1. ユーザインタフェースに親しむ 17
3.2. タブ操作 19
チュートリアル１：Vehicle Spy の基礎 21
パート１：Logon とロード 22
パート２：メッセージトラフィックの観測 23
パート３：メッセージの詳細を確認 24
パート４：実行中にメッセージの詳細をチェック 25
パート５：”Module Message”のみを表示(フィルタリング) 26
パート６：その他のフィルタ 28
パート７：メッセージのタイムスタンプ 29
パート８：メッセージビューバッファの保存 30
パート９：メッセージビューバッファの消去 31
例題 32
チュートリアル２：メッセージ送信 33
パート１：セットアップファイルのロード 33
パート２：送信メッセージの入力 34
パート３： 代替ワークスペースの活用 34
パート４：メッセージの手動送信 35
パート５：メッセージの周期的送信 36

パート６：メッセージへの自動応答 37
例題 39
チュートリアル３：メッセージからシグナルをデコード 40
パート１：設定ファイルのロード 40
パート２：メッセージ入力 40
パート３：アナログ値に対して式を入力 41
パート４：メッセージデータ表示 43
パート５：Signal List と Signal Plot からメッセージデータを観る 44
パート６：Signal Plot 入門 45
例題 47
チュートリアル 4A：キャプチャ・ファンクションブロックとグラフィカルパネル 48
パート１：シミュレーションの設定 48
パート２：キャプチャ・ファンクションブロックの設定 50
パート３：グラフィカルパネル 52
パート４：まとめ 56
チュートリアル 4B：スクリプト及びプレイバック・ファンクションブロック 56
パート１：アプリケーションシグナル 57
パート２：グラフィカルパネル 58
パート３：プレイバック・ファンクションブロック 60
パート４：スクリプト・ファンクションブロック 61
パート５：まとめ 64
チュートリアル５：DTC データのリクエスト及び保存 65
パート１：診断ジョブを作成 66
パート２：データを要求及び保存するファンクションブロックを作成 66
パート３：グラフィカルパネルを作成 67
パート４：DTC の要求、保存、表示 68
パート５：まとめ 68
チュートリアル６：DPS プログラミングの自動化 69
パート１：DPS プログラミングジョブを作成 69
パート２：ジョブを起動するファンクションブロックを作成 70
パート３：グラフィカルパネルを作成 71
パート４：DPS プログラミングパネルを利用 72
パート５：まとめ 72
チュートリアル７：診断機能の応用 73
パート１： DID セットアップの読込 73
パート２： DID セットアップの書込 74

パート３： Read DID の書込 74
パート４： ダイナミック DPID からの PIDS のリクエスト 75
パート５： ダイナミック DPID の要求 76
パート６： まとめ 76
チュートリアル８：CoreMini の使用と設定 77
パート１：送信メッセージの作成 77
パート２： データロギング 78
パート３： スクリプトのロード 79
パート４： スクリプトの実行 81
パート５：データの抽出 81
パート６：データの表示 82
パート７：まとめ 82
チュートリアル９：C Code Interface 概要 83
パート１：プロジェクト作成 83
パート２：イベントの追加 84
パート３：サンプルの実行 85
パート４：デバッグ 85
パート５： まとめ 85
3.3. チュートリアルまとめ 86
3.4. チュートリアル用ファイル 86

1. Welcome!



本ドキュメントは、あらゆるレベルのユーザに向けて作成されました。

特に、新規もしくは将来のユーザの皆様へ：

もし Vehicle Spy を入手されたばかり、もしくは単に本製品に関して情報収集をされているのであれば、2 章をまず御覧ください。Vehicle Spy の要点、多くの使用事例、仕様や要件について、簡単に把握できます。

Vehicle Spy の基礎を学ぶ上で、チュートリアルをご覧頂くのもお勧めです。それが完了すれば、Vehicle Spy を実際に利用する上での、確かな知識が得られるでしょう。

全てのユーザ皆様へ：

Vehicle Spy の機能詳細に関しては、チュートリアル以降の文書をご覧下さい。

もし久々に Vehicle Spy を使用される場合は、再びチュートリアルに戻るのもお勧めです。

また他の製品に関しての情報が必要であれば、12 章をご覧下さい。Intrepid Control Systems, Inc.社の利用可能なハードウェアに関する情報が得られます。

2. イントロダクション

2.1. 概観

ネットワーク搭載車両のための車載エレクトロニクス・エンジニアリング・ツール

Vehicle Spy(ビークル・スパイ)は、現在、また将来の車載ネットワークのデザイン・テスト・解析のための、世界トップクラスのエンジニアリングツールです。

自動車のエレクトロニクスの複雑さは、１０年前の数えるばかりの電子機器から、ネットワークに接続された何十台もの機器へと、前例のないレベルへと発展してきています。

さらなる複雑化の要因として、ゲートウェイで接続されたメイン・ネットワーク、サブ・ネットワーク、マルチメディア・ネットワークを含む複合ネットワークが挙げられます。この新たな複雑性が、車載エレクトロニクス・システムの開発と検証の難しさに拍車をかけています。

Vehicle Spy は、使いやすさと生産性にフォーカスし、そのような困難の解決に貢献します。Vehicle Spy には、情報への迅速なアクセスを支援する沢山のツールが含まれています。

VehicleSpy の使用例

- **バス解析** – すべてのネットワークメッセージを同時に観測
- **フライトレコーダ** – バスイベントを自動的にキャプチャし、バッファファイルを保存
- **ノードシミュレータ** – 欠けているノードやゲートウェイ、もしくは車両全体をシミュレーション
- **フラッシュ / 診断ツール** – ECU ソフトウェアのフラッシュ、メモリの読込/書込、エラーコードの読込

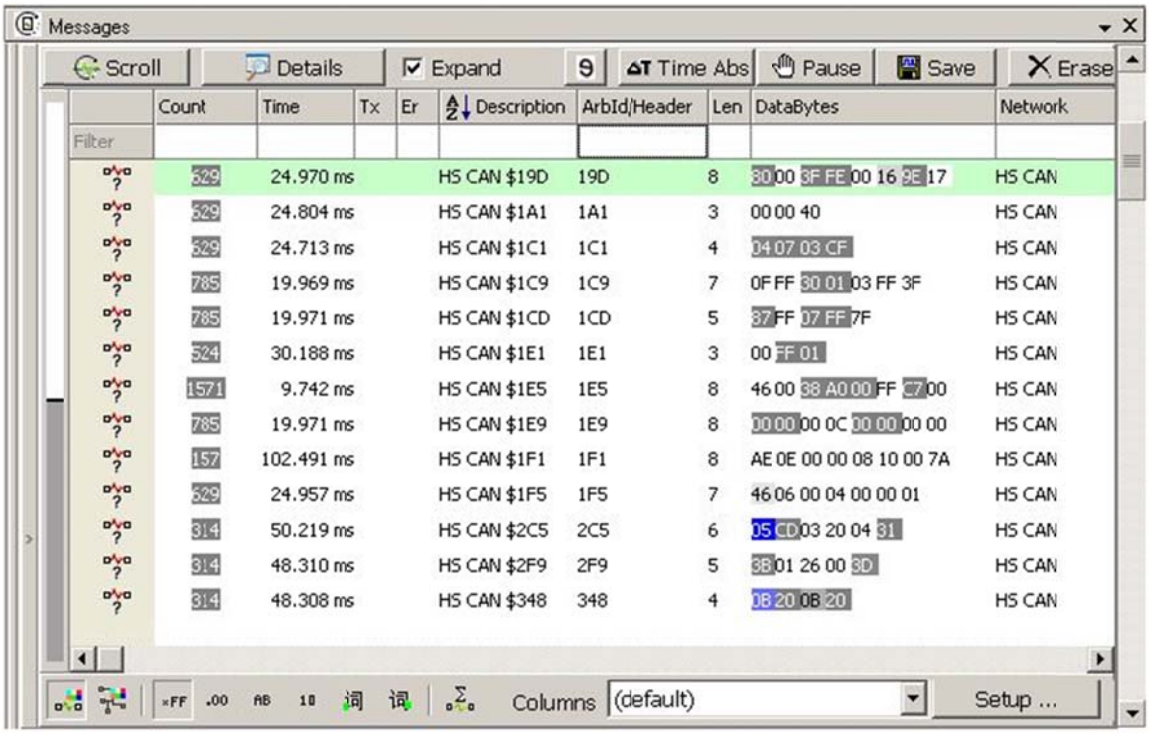
同時に複数プロトコルをサポート

CAN（デュアル & シングル・ワイヤ）, Ford SCP, Ford UBP, GM CGI, GME CIM SCL K-line, GMLAN, ISO9141, J1708, J1850 PWM, J1850 VPW（GM Class 2 & Chrysler 1 バイト・ヘッダ）, J1857, J1939, Keyword 2000, LIN, LSFT CAN, UART

2.2. 要点

Vehicle Spy が提供する主な機能を、要点に絞って説明します。

2.2.1. バストラフィック観測



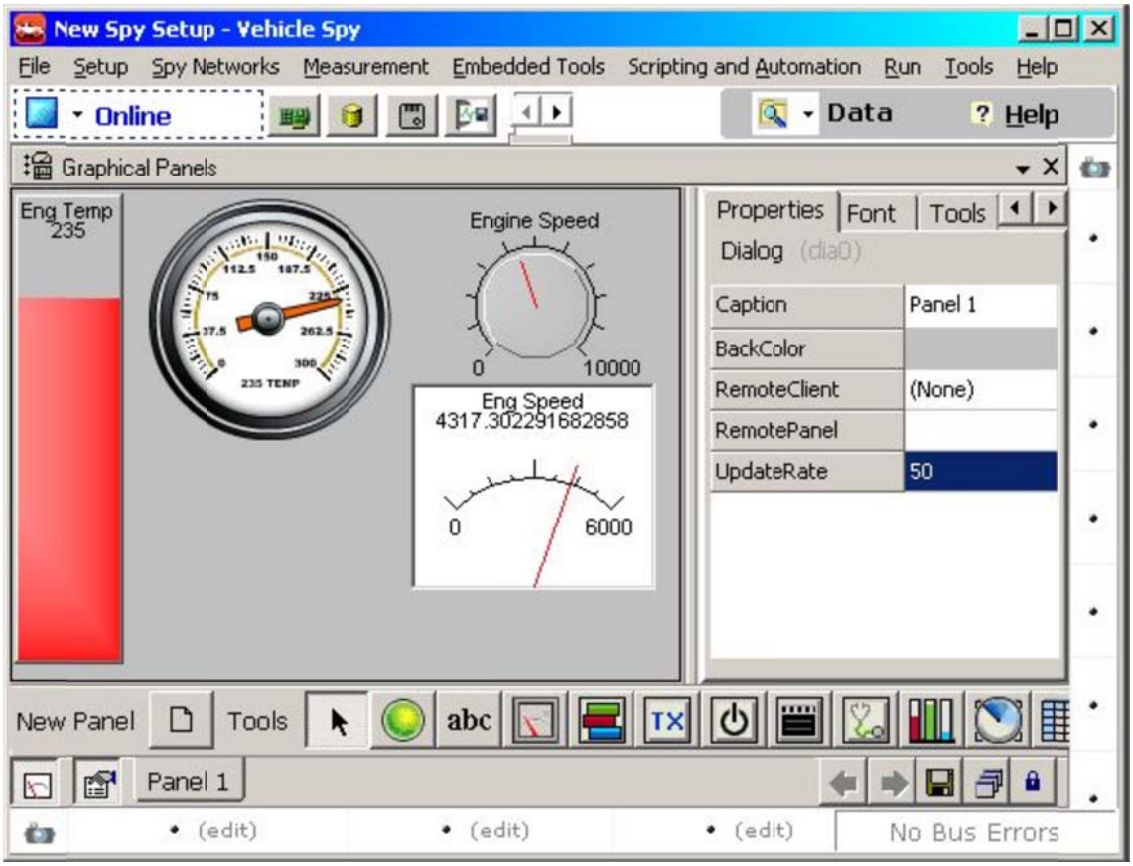
ネットワークに現れるメッセージの観測には、バストラフィック観測機能が使用されます。Vehicle Spy には、メッセージ表示の方法として、固定表示とスクロール表示があります。これらの表示は、ボタンのクリック 1 つで切り替え可能です。メッセージには、タイムスタンプ及びカウント数が付随します。

メッセージの観測に加えて、Vehicle Spy は他にも多くの有用なバス観測機能を提供します。1 つは、フィルタリング機能です。特定のバイトを持つメッセージに対するフィルタリングは、ユーザが興味あるメッセージに集中する上で非常に有用です。例えば、あるクラスタからのメッセージだけを見たい場合、そのクラスタからの ID バイトを持つ特定のメッセージに対して、フィルタリングができます。フィルタの作成は、メッセージ・データベースからフィルタの仕様を直接入力することで、容易に可能です。

Vehicle Spy は、メッセージに関するテキスト形式の記述と、メッセージに含まれるシグナルのデコードによって、メッセージの識別機能をさらに強化します。また色分け機能も、メッセージを即座に認識する上で有効です。

バストラフィック観測機能を利用する場合、通常、メッセージはバッファに収集され、保存されます。効率化のため、Vehicle Spy においては分析作業の自動化が可能です。Vehicle Spy は、バッファの統計、収集後のフィルタリング等、有用な機能を提供します。メッセージバッファは、CSV（Comma Separated Value）ファイルとして保存され、そのファイルは解析に有用な Microsoft Excel と完全に互換です。

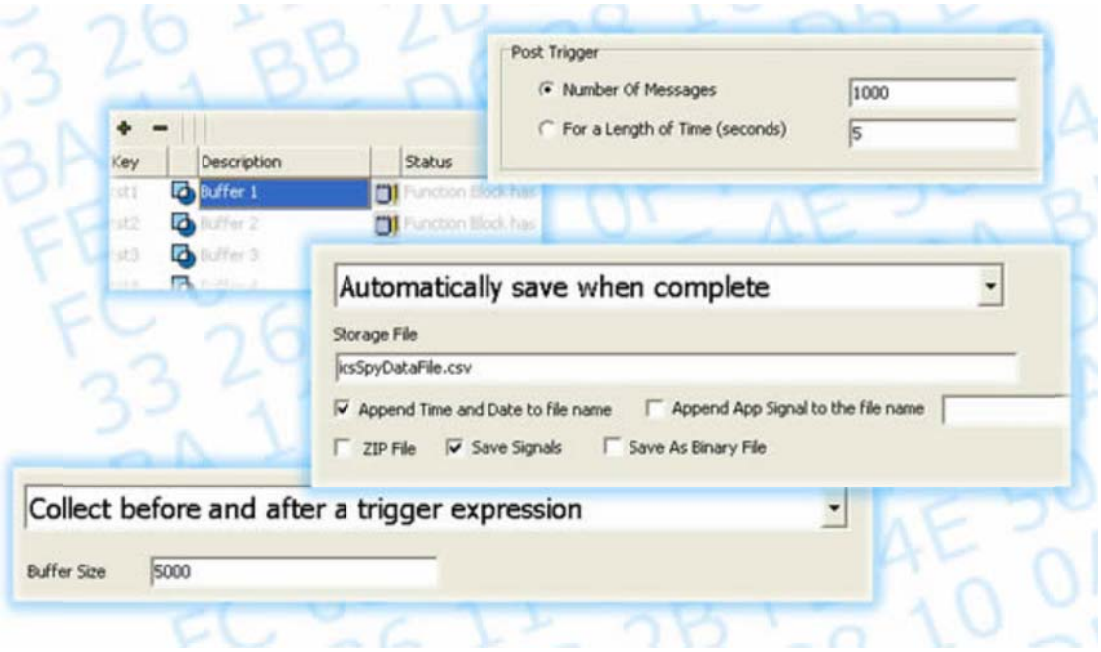
2.2.2. シグナル観測



シグナルの観測と保存とは、車両上にセンサ等の配線をせずに、RPM、冷却水温度、エンジン負荷、スロットル位置、MAP のようなデータを収集することを意味します。これにより、車両データを収集する人の時間と費用が節約できます。

車載ネットワークからのデータ収集に加えて、Vehicle Spy は車載ネットワークデータと共に、センサからのデータも収集可能です。この機能は、バスデータを物理データと比較する多くの状況で、非常に有用です。

2.2.3. フライトレコーダ



フライトレコーディングとは、ある時間内に発生する問題を観測・キャプチャするために、車載ネットワークを連続的に観測することです。ネットワークメッセージの収集と保存は、柔軟なトリガ条件に基づいてメッセージバッファをキャプチャできる、デジタル・オシロスコープに非常に似ています。例えば、エンジン回転数が 5000 を超える前後の、1000 メッセージを収集することが可能です。

同時に複数のバッファを収集できるため、複数の問題が観測されている場合にも対応可能です。フライトレコーディングのもう 1 つの便利な利用例は、GPS（Global Positioning System）で、データ収集時に使用されたテストコースと、車載データの協調を可能とします。

2.2.4. ノード・車両シミュレーション

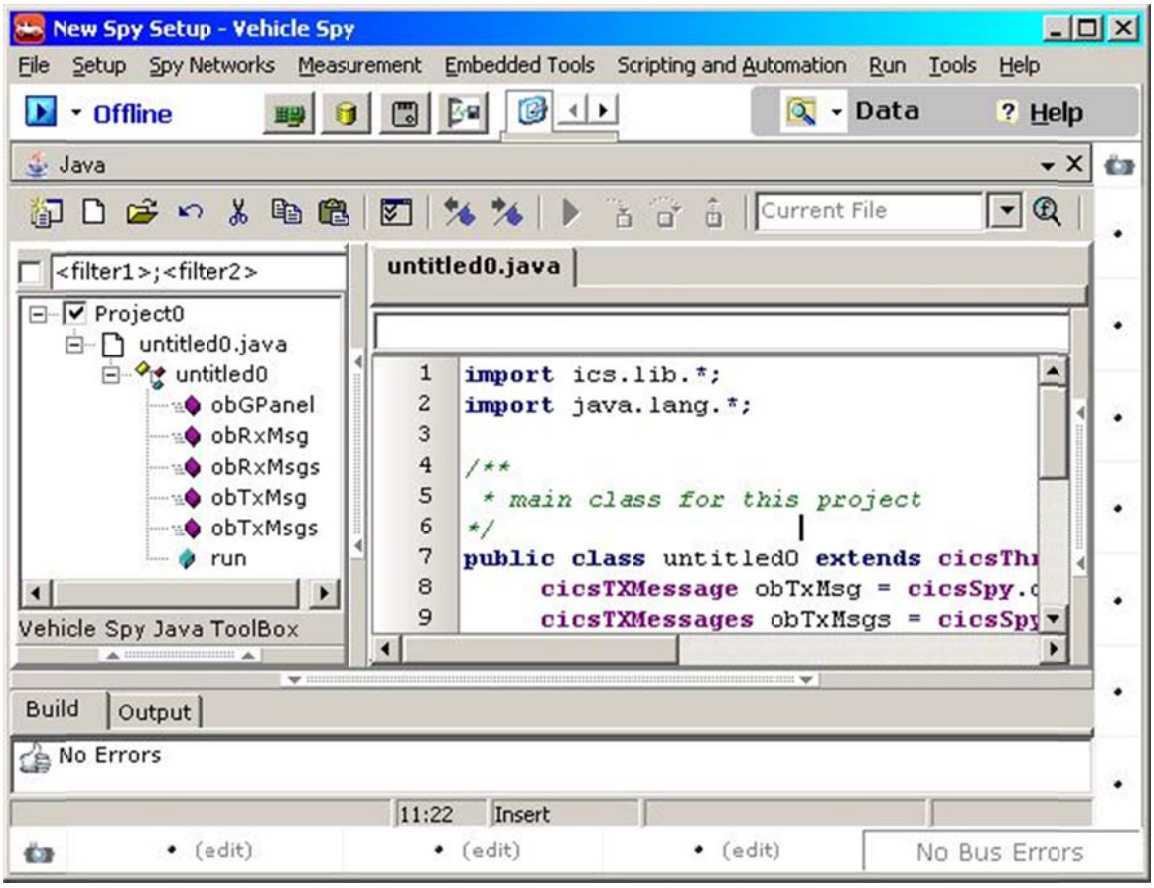
ノード・車両シミュレーションは、ネットワーク上の電子モジュールまたは車両をシミュレーションします。開発エンジニアは、ネットワーク・ソフトウェア・インターフェースを開発する必要がありますが、ネットワークまたは車両を持っていないことが多々あります。通信先のモジュールが、利用可能でないかもしれません。Vehicle Spy のシミュレーション機能を使用して、エンジニアはそれらのモジュールなしでテストを実現可能です。

ノードシミュレーションにおいては、受信メッセージに対する応答が頻繁に行われます。これは、モジュールが特定のメッセージを受信すると、特定の時間内にメッセージで応答しなければならないことを意味します。この応答は、受信メッセージの関数となることもあります。例えば、受信メッセージ中のビットが、いくらか改変されて、応答メッセージとして返される場合です。

Vehicle Spy のプレイバック機能によって、車両トラフィックのシミュレーションが可能です。例えば、シミュレーション用データは、車両からの収集・保存が可能です。その後、そのデータをモジュールに対してプレイバックすることが可能です。これにより、車両からキャプチャされた障害を再現し、テストを実行できます。

Vehicle Spy はメッセージを、周期的にまたは特定イベントが発生したときに、生成することも可能です。これらの機能は、ノード・車両シミュレーションで頻繁に役に立ちます。

2.2.5. テスト自動化



テスト自動化とは、電子モジュールのテストを自動的に行うことです。これは、メッセージの送受信、バッファのキャプチャ、受信データへの応答の自動化を含みます。Vehicle Spy は、そのような自動化作業を簡単にするツールを有しています。 1 つは、ファンクションブロックと呼ばれるツールです。これにより、簡単なマウス操作でのスクリプト作成とデータキャプチャが可能です。もう 1 つのツールは Java です。Vehicle Spy には、Java エディタ、及び Java コードをシンプルかつ強力に扱う機能が組み込まれています。また、C コードによるスクリプト作成もサポートしています。

2.2.6. まとめ

Vehicle Spy は、電子モジュール開発、診断ツール開発、検証テスト、車載耐久性テスト、テスト・コンサルティング、車載データ収集、等の担当者に対して、大きなメリットを提供します。以下の表に概要を示します。

各担当者に対するメリット						
使用例	電子モジュール開発	診断ツール開発	検証テスト	車載耐久性テスト	テストコンサルティング	車載データ収集
1) バストラフィック観測	高	高	高	中	低	低
2) 値の記録	低	高	中	中	低	高
3) フライトレコーダ	中	中	中	高	低	低
4) ノードシミュレーション	高	高	中	低	低	低
5) テスト自動化	低	低	高	低	高	低

2.3. 世界のユーザ様一覧

Vehicle Spy は、世界の主要な自動車メーカーやサプライヤに、車載ネットワークの診断・検証・開発・テスト用ツールとして、採用頂いています。ユーザとしては、General Motors、Visteon、Delphi Delco Electronics、Valeo、Lear Corporation、TRW、Ascent Technologies、Motorola、Panasonic、Continental-Teves、SPX Corporation などが挙げられます。詳細は弊社 HP を御覧ください。

2.4. 仕様

Vehicle Spy アプリケーション	複数の車両ネットワークに対する、同時観測・送信が可能。異なる複数の車両ネットワーク・プロトコルに対しても対応可能
	カスタム・タブ機能によって、効率的な画面操作が可能
	Vehicle Spy 設定ファイルを保存、呼び出し
	Logon 機能により複数ユーザーを管理、直近の 100 のセットアップ・ファイルの履歴を保持、またデータ・ディレクトリを管理
	Java スクリプトによるカスタム機能の生成
	丁寧なチュートリアルを含むオンライン・ヘルプ
	GPS（Global Positioning System）の利用
	リバースエンジニアリングに適したメッセージ検出機能
メッセージビュー	メッセージに関するテキスト形式の記述を表示
	RPM、ペダル位置、速度等のメッセージ中の全てのシグナルを表示
	ネットワーク、ノード、メッセージ、データ・タイプに基づく包含/除外フィルタ
	CAN と J1850 のデータを同じ画面に表示
	観測を一時停止し、過去のデータを表示
	絶対/相対時間表示
	Excel 互換フォーマットで、メッセージ記述とシグナルを含むデータファイルの保存及び表示
	データ収集後のフィルタリングと解析
	16 進、10 進、2 進、ASCII データ表示
	メッセージ、ネットワーク、ノードに基づく色分け
	バッファ容量は、システム・メモリに依存。百万メッセージ以上の履歴保持が可能
	フォントと色のカスタマイズ
	バス負荷の表示
	変化量による強調：現在観測されていないメッセージやシグナルは、黒から灰色となりフェードアウト
	データバイトの変化による強調：データバイトの変化時に、青色へと変化。メッセージの状態変化の検出に有用
	メッセージ中の任意の情報に基づく、メッセージ表示の整列
	右クリックメニューから、メッセージをメッセージエディタへと簡単にコピー
メッセージエディタ	それぞれ別個の表に設置された受信メッセージと送信メッセージ
	オンライン時の動的編集(Vehicle Spy を停止・オフライン化することなく、メッセージの編集が可能！)
	バイト、ニブル（４ビット）、ビット単位でのメッセージの特定
	データベースボタンの押下で、データベースからのメッセージやシグナルの簡単な呼び出し
	メッセージから、任意の式を用いてシグナルを算出（デジタル、ステート・エンコーデッド、テキスト、アナログ）
	カスタム可能なデジタル及びアナログ値のフォーマット（２進数、１６進数、物理値、数学的表記）
	任意のネットワーク上の、送信メッセージとのリンクにより、ネットワークの変化に基づく自動応答・送信を実現
	計算シグナル型を利用した、正弦関数、ランプやカスタム関数によるシグナル生成

	メッセージ長を Expected Length Input 機能により検証
	エラーやバスイベント等のメッセージの特徴による特定
	メッセージごとの色分け機能
	メッセージとシグナルの切り取り、コピー、貼り付け、整列
送信パネル	メッセージエディタに定義された送信メッセージは、自動的に送信パネルに表示
	送信メッセージは、受信メッセージまたはシグナルから算出が可能
	周期的メッセージ送信
	手動メッセージ送信ボタン
グラフィカルパネル	表示器やコントローラを持つアプリケーションを自作
	表示器やコントローラの内容：グラフ、送信ボタン、ファンクションブロックボタン、On/Off ボタン、ノブ、メーター、LED/ライト、テキスト表示、テキスト入力、数値入力、グラフィカルディスプレイ、ドロップダウンリスト、気圧計、グリッド、アクションボタン、スクローリングモニタ
	グラフは X 軸 vs Y 軸、または Y 軸 vs T 軸
	PRNDL や電源モードのような、ステート・エンコーデッドパラメータに対するドロップダウンコントロール
	アクションボタンによるパネルのプリント、パネルの切り替え、ユーザのデータディレクトリへのアクセス
	グラフィカルディスプレイによる、パネル上のロゴ等、色付きの線や図の作成
	カスタムに便利な機能を多く含む各種ユーザインタフェースパーツ
	Visual Basic スタイルのプロパティエディタ
ネットワーク	任意の数のネットワークをサポート
	ネットワークごとの色分け
	CAN と J1850 ネットワークのためのネットワーク統計：メッセージカウント、メッセージレート、メッセージレート最大値、バス占有率、バス占有率最大値
ノード	任意の数のノード数をサポート
	異なるネットワークからのノードをサポート
	ノード毎の色分け
	CAN と J1850 ネットワークのためのノード統計：メッセージカウント、メッセージレート、メッセージレート最大値、バス占有率、バス占有率最大値
ファンクションブロック	キャプチャ済みネットワークトラフィックに、任意のフィルタを追加してリプレイ
	次のスクリプトコマンドを有するカスタムスクリプトの作成：ジャンプ、メッセージ送信、一定時間の待機、ある式が真になるまで待機、別のファンクションブロック、受信メッセージに値や式を代入、メッセージ統計をリセット、スクリプト停止、グラフィカルパネルを表示、ある式が真ならばジャンプ、等
	任意の式に基づく、トリガ前後のネットワークバッファキャプチャ
	ユニークなファイル命名法とともに、ネットワークトラフィックを保存
	イベントに基づくデジタル出力駆動
	メッセージ間の遅延やオフセットを調整して、メッセージ群を送信
	式・Java・他のファンクションブロックに基づく、ファンクションブロックの開始・停止・トリガ
	任意の数のファンクションブロック生成

データベース	車載ネットワーク上にて送受信されるメッセージを定義したデータベースの読込
	データベースによるメッセージとシグナルの自動記述・表示
	メッセージエディタへの迅速な入力
	インポート・エクスポートのための、DBC と UEF ファイルフォーマットサポート
	各ネットワークがそれぞれ別個のデータベースを保持可能
診断	診断リクエストの設定及び起動
	診断応答のデコードのためのデータベース使用
	データは、可読性の高いテキスト形式と、生データメッセージにて表示

2.5. 要件

PC システム要件

最低条件

- Windows XP (SP 3), Windows Vista (Pre SP, SP 1, or SP2) or Windows 7 (Pre SP or SP1)
- RAM : 256 MB
- CPU : Intel Celeron 1 GHz
- 画面解像度：1024 x 768
- neoVI (Blue, Yellow, Red, FIRE), もしくは ValueCAN

推奨条件

- Windows XP (SP 3), Windows Vista, or Windows 7
- RAM: 1GB 以上
- CPU: 2.1 GHz 以上
- 画面解像度: 1280 x 1024 以上
- neoVI (Blue, Yellow, Red, FIRE), もしくは ValueCAN

オプション

- オーディオカード
- GPS ユニット（NMEA RS232 Interface）
- PC ジョイスティック

車両ネットワーク・インターフェース・ハードウェア

Vehicle Spy は、Intrepid Control Systems, Inc.社製の、車載ネットワークのためのハードウェア・インターフェースをサポートしています。これらは、Vehicle Spy と密に統合されています。

neoVI FIRE 車両インターフェース

推奨：CAN, J1850, GMLAN, GM Class 2, ISO9141, Keyword 2000, OBD-II, LIN, J1939

neoVI BLUE 車両インターフェース

推奨：CAN, J1850, GMLAN, GM Class2, ISO9141, Keyword 2000, Ford SCP, OBD-II, J1587, J1708, LIN, J1939

neoVI YELLOW 車両インターフェース

推奨：DW CAN, J1850, GM Class 2,ISO9141, Keyword 2000, J1587, J1708, LIN, J1939

neoVI RED 車両インターフェース

推奨：DW CAN, ISO9141, Keyword 2000, LIN, J1939

ValueCAN3 車両インターフェース

推奨：DW CAN, J1939

2.6. ウェブサイト

Vehicle Spy の詳細に関して、弊社 HP もご覧下さい。

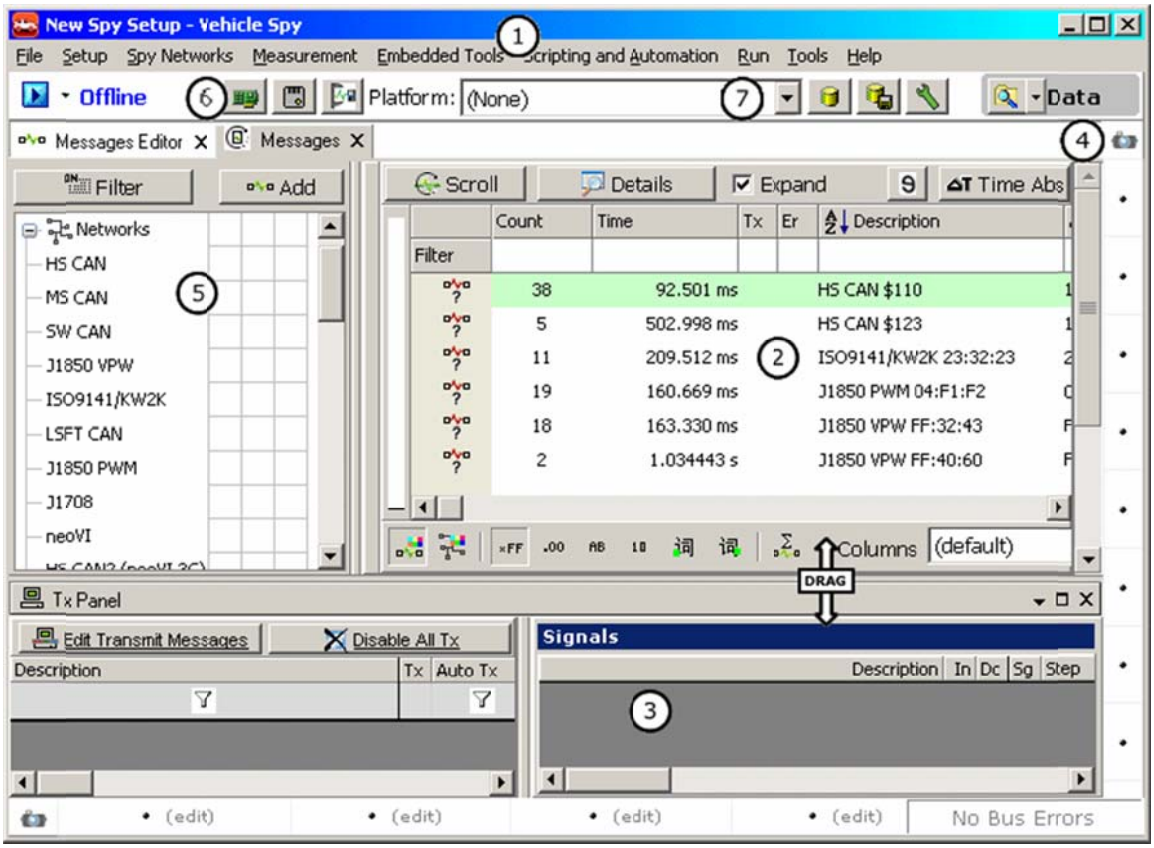
3. チュートリアル

本チュートリアルは、Vehicle Spy の学習を迅速かつ容易に行うために作成されました。現在、利用可能なチュートリアル一覧は以下のとおりです。

3.1. ユーザインタフェースに親しむ

実際のレッスンを始める前に、Vehicle Spy のインタフェースに慣れることが重要です。

Vehicle Spy のインタフェースは、下図の様に複数のパネルに分割されます。



①**メニューバー**：Vehicle Spy のインタフェースの上部には、標準的なメニューバーがあります。これには、File, Setup, Spy Networks, Measurement, Embedded Tools, Scripting and Automation, Run, Tools, Help が含まれます。メニューバーから、Vehicle Spy のすべての機能へとアクセスが可能です。メニューバーには、Data ボタンも含まれます。このボタンにより、ユーザのデータディレクトリへの迅速なアクセスが可能です。また、Vehicle Spy で作業を進めるにつれ、メニューバー中に複数のタブが現れることになるでしょう。それ

らのタブは、メインワークスペース中に表示されるビュー[1]をコントロールするために使用されます。メニューバーもしくはビューアイコン(画面右上もしくは左下のカメラ型アイコン)からビューを選択すると、そのビュー用のタブが、自動的にメニューバー上に現れます（上図では、Messages ビューのタブが現れています）。タブを削除する場合は、タブの右側の×をクリックします。このようなカスタマイズ可能なタブにより、個々のユーザが、自身にとって最適なインタフェースを作成できます。

②メインワークスペースパネル：Vehicle Spy ユーザインタフェースの大部分は、メインワークスペースパネルが占めています。Vehicle Spy を起動した際に現れる Logon 画面など、殆どの操作がこのワークスペース上で行われます。メインワークスペース上に現れるビューは、ビューの選択に基づき変化します。別のビューを選択するには、まずビューのグループ（Vehicle Networks, Measurement, Embedded Tools, Scripting and Automation）を、メニューバーもしくはカメラ型のビューアイコンから選択し、その中からビューを選びます。

③ & ④代替ワークスペースパネル：しばしば、複数のビューを同時に見る必要があります。これに対応するために、Vehicle Spy は代替ワークスペースパネルを用意しています。それらのパネルは、Vehicle Spy インタフェースの下側及び右側に格納されており、カスタマイズ可能なタブのセット等、メインワークスペースと同等の機能を有します。

⑤フィルタパネル：メインワークスペースの左には、フィルタパネルがあります。このパネルは、Messages ビューが選択されたときのみ有効であり、Vehicle Spy の多様なフィルタ機能を有効化する際に使用します。

⑥クイック設定ボタン：あるプロジェクトから別のプロジェクトへと切り替える際には、ネットワークとハードウェアの設定変更が必要となる場合があります。クイック設定ボタンは、それらの共通のビューへのショートカットです。

⑦プラットフォームの選択と設定ボタン：このドロップダウンメニューは、全ての設定済みのデータベースと、データベースオプションへのショートカットです。

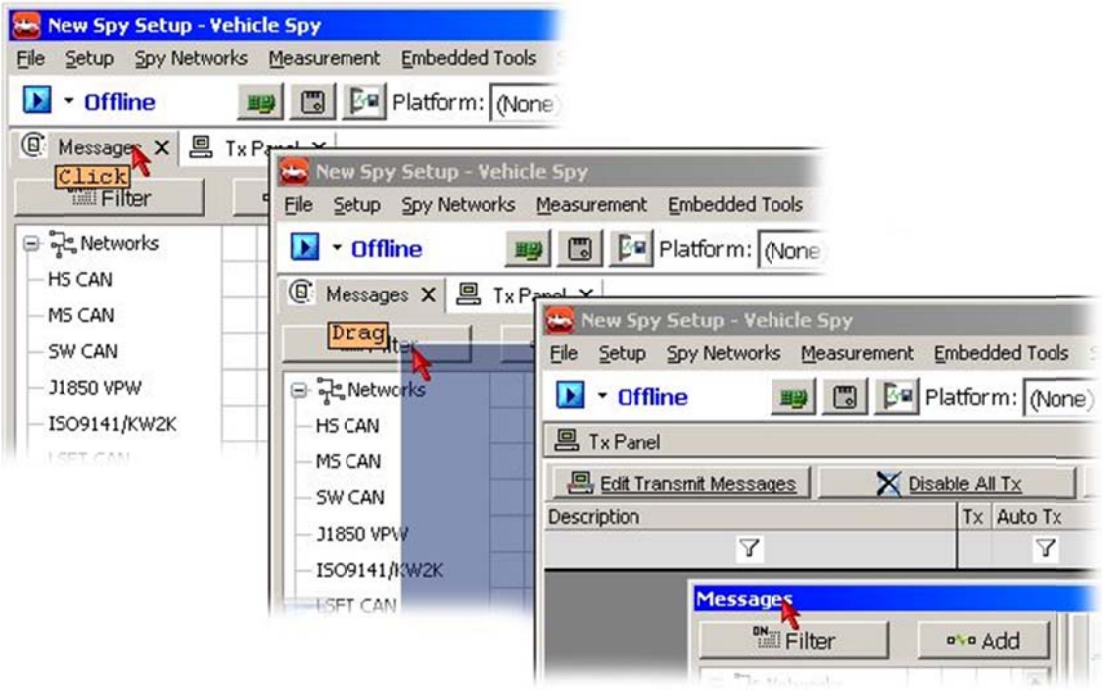
これらのパネルのすべては、ユーザの好みに応じて、開く、閉じる、移動、サイズ変更、さらにはメインウィンドウからの分離、といった動作が可能です。サイズ変更には、マウスのポインタが対の矢印に変わるまで、パネル間の線へとマウスを移動します。そこをクリック、ドラッグし、パネルが希望のサイズになったときに離します。上図の マークを御覧ください。

[1] 訳注：メッセージエディタ、送信パネル等の独立した各種パネル

3.2. タブ操作

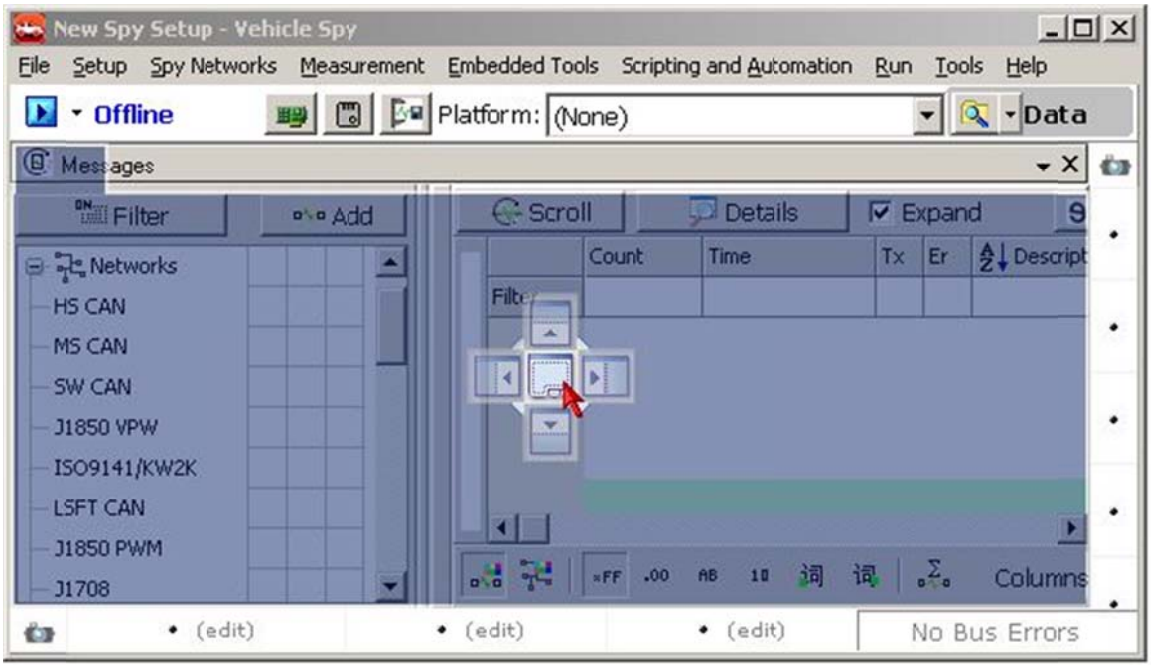
作業効率向上のため、Vehicle Spy はメインワークスペースからタブを分離する機能を備えています。この機能により、ビューを最も適切な場所に設置することが可能です。

タブの分離（下図）： Vehicle Spy のタブは、メインウィンドウ[2]から分離が可能です。これは、同時に沢山のビューをみる必要があるとき、もしくは画面上の異なる場所に異なるウィンドウを配置したいとき、有用です。クリック＆ドラッグにより、メインウィンドウから分離します。



分離したタブの再配置（下図）：分離したタブは、元に戻すか、他のビューへの設置が可能です。これは、ビューのタイトルバーをクリック＆ドラッグし、結合先のビューへと移動することで行います。移動中のビューが他のビューへと近づいた時、結合に関するオプションが現れます。マウスをそのオプション上へと動かすと、結合先のビュー上に、移動予定先の領域がハイライトされ、マウスを離すことで、その場所へと設置されます。ビューの結合を行わない場合は、キーボード上の Ctrl キーを押しながらドラッグ操作を行います。

[2] 訳注：Vehicle Spy の、Windows アプリケーションとしての全体のウィンドウ


New Spy Setup - Vehicle Spy
File Setup Spy Networks Measurement Embedded Tools Scripting and Automation Run Tools Help
Offline
Platform: (None)
Data
Messages
Filter
Add
Scroll
Details
Expand
Networks
HS CAN
MS CAN
SW CAN
J1850 VPW
ISO9141/KW2K
LSFT CAN
J1850 PWM
J1708
Count
Time
Tx
Er
Descript
Columns
(edit)
(edit)
(edit)
No Bus Errors

チュートリアル１：Vehicle Spy の基礎

概要

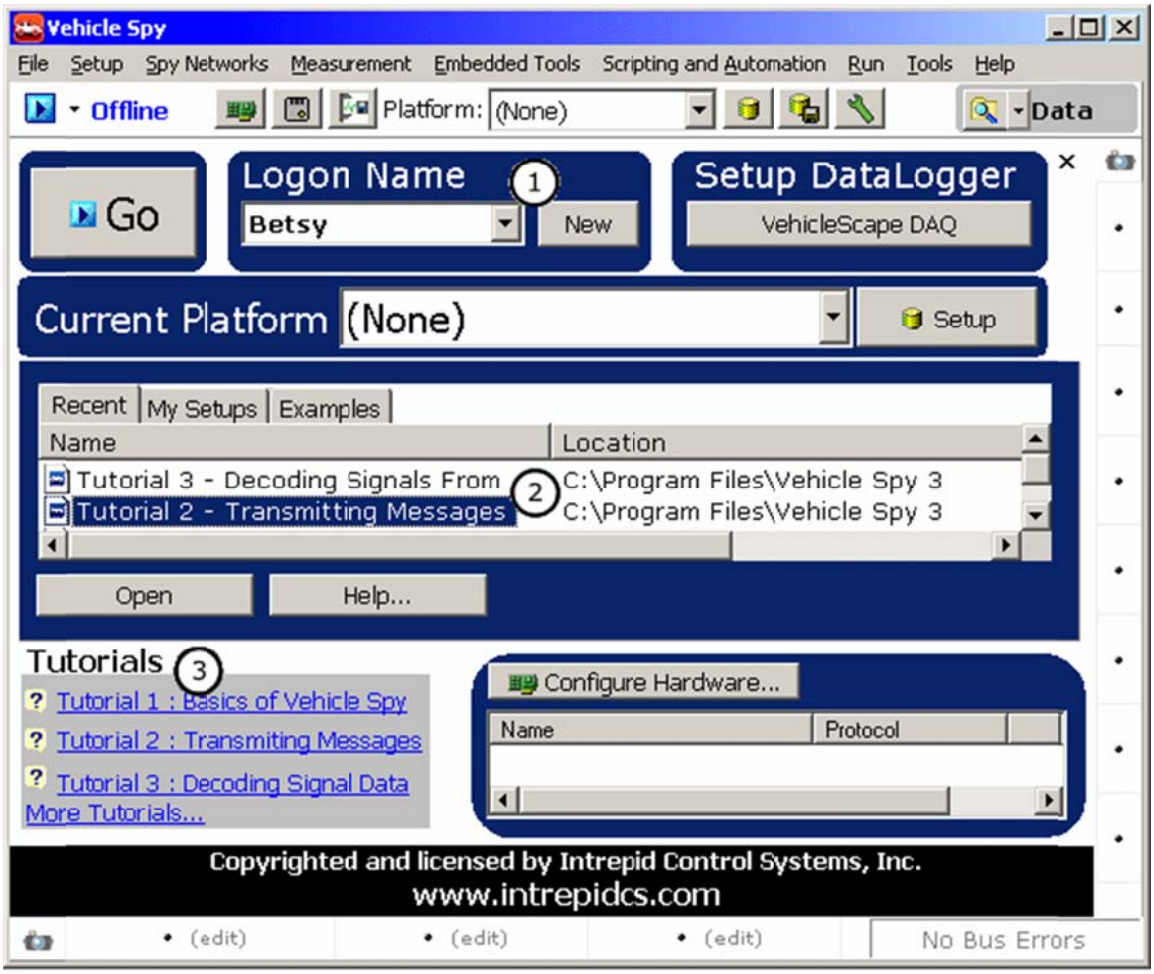
この最初のチュートリアルは、Vehicle Spy に慣れる上で非常に有用です。本課題を通じて、Logon、メッセージトラフィックの表示、メッセージフィルタリング等の、沢山の重要なスキルを学ぶことができます。

Good Luck！

パート１：Logon とロード

Vehicle Spy のインタフェースやビューメニューの操作方法に慣れてくれれば、本チュートリアルを始める準備は完了です。以下の例に倣って注意深く進めてゆけば、すぐに Vehicle Spy の達人へとなれるでしょう。

1) **Vehicle Spy の起動：**Vehicle Spy のインストール後、デスクトップにアイコンができます。そのアイコンをダブルクリックし、Vehicle Spy を起動します。デフォルトで、下図の Logon 画面が開きます。Vehicle Spy をすでに起動しているか、Logon 画面をスキップした場合、メニュー・バーの File から Logon を選択することで、Logon 画面が開きます。



2) **Logon Name リストから Logon Name の選択：**Vehicle Spy は、データファイル及び設定ファイルの、各ユーザ固有のデフォルトの場所を設定するために、Logon 名を使用します。以前に同じ PC 上の Vehicle Spy にログインした場合は、ドロップダウンリストからその名前を選択して下さい。新規ユーザの場合、Logon 名を新規作成する必要があります。Logon Name 欄の横の New ボタンをクリックして下さい（①）。ユーザ名もしくは好みの名称を入力して、OK をクリックします。

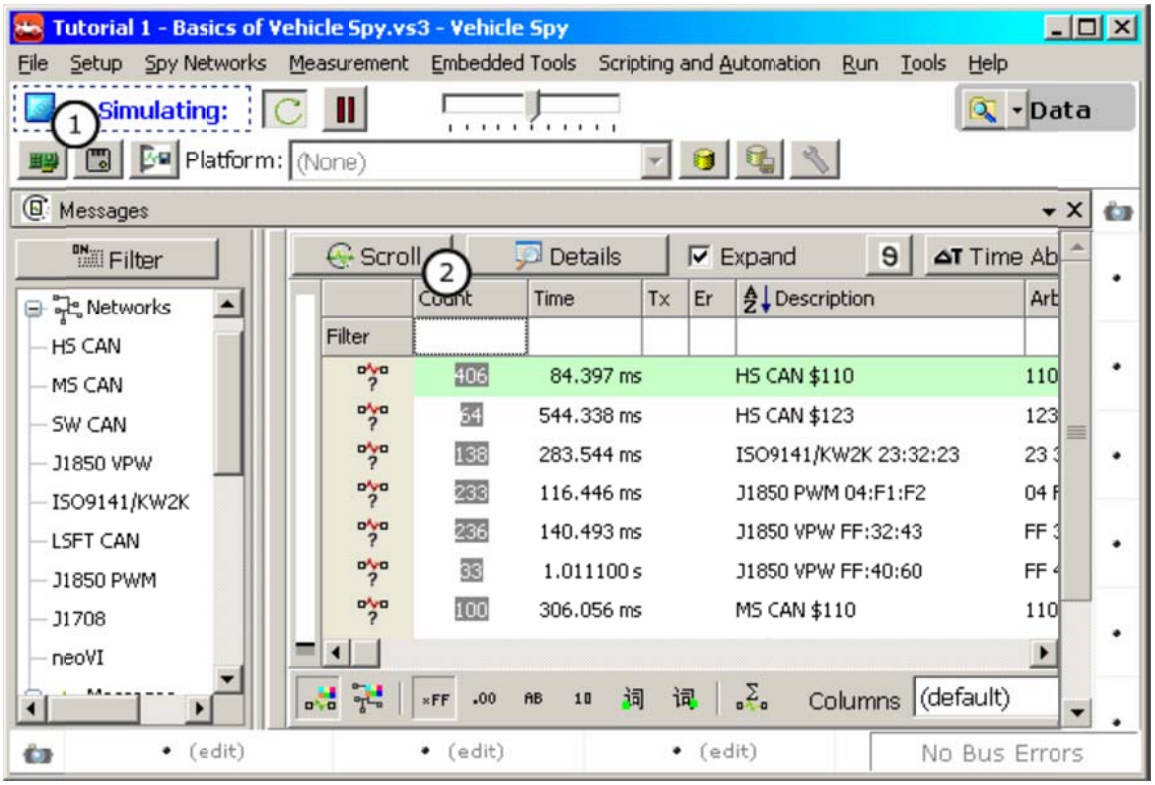
3) **Vehicle Spy 設定ファイルの読込：**本チュートリアルを開始するためには、”Tutorial 1 – Basics of Vehicle Spy”という Vehicle Spy 設定ファイルを読み込む必要があります。このファイルは、チュートリアル 1 の全ステップの完了に必要なすべてのデータが含まれています。この設定ファイルをロードする

ためには、Logon ビューの Examples タブを選択し、一覧からファイルを選択します（②）。ファイルを選択後、Open をクリック（③）することで、設定ファイルがロードされます。（メニューバーの上部に”Tutorial 1 – Basics of Vehicle Spy”と表示されることを確認下さい）

パート２：メッセージトラフィックの観測

1) **Messages ビューの選択：**メニュー・バーを使用して、Vehicle Networks、続いて Messages を選択すると、メインワークスペースに、Messages ビューが開きます。一度このビューを選択すると、Messages ビュー用のタブが表示されるようになります。Vehicle Spy は常に、選択したビューに対応したタブを生成します。Vehicle Spy は、個々のユーザに応じたインタフェースを作成することで、可能な限り効率的な作業を実現するようデザインされています。

2) **Vehicle Spy 開始：**インタフェースの左上隅に位置する青いスタートボタン（①）をクリックします。直ちに、変化するデータがメインワークスペース中に現れ、またスタートボタンがストップボタンへと切り替わるはずです。



3) **メッセージトラフィックの観測：**画面中のデータは、過去に記録されたものです。ここでは単にチュートリアルのため、それを再生しています。Vehicle Spy においては、このような過去のデータの再生をシミュレーションモードと呼びます。このモードは、スタートボタン横の青い文字列で示されています。シミュレーションモードは、ハードウェアや実車両への接続なしに各種機能の試用や実験を可能とし、特にトレーニング目的において非常に便利なオプションです。

Messages ビューには、スクロールビューと固定ビューの二種類があります。Messages パネルの左上にある Scroll ボタン（②）をクリックすることで、これらを切り替えます。

Scroll ボタンを押すと、リアルタイムに流れてゆくメッセージを観測できます。このパネル上部の Pause ボタンをクリックすると、Vehicle Spy を停止せずに、スクロールを停止します。右側に現れるバーにより、手動でのメッセージのスクロールが可能です。

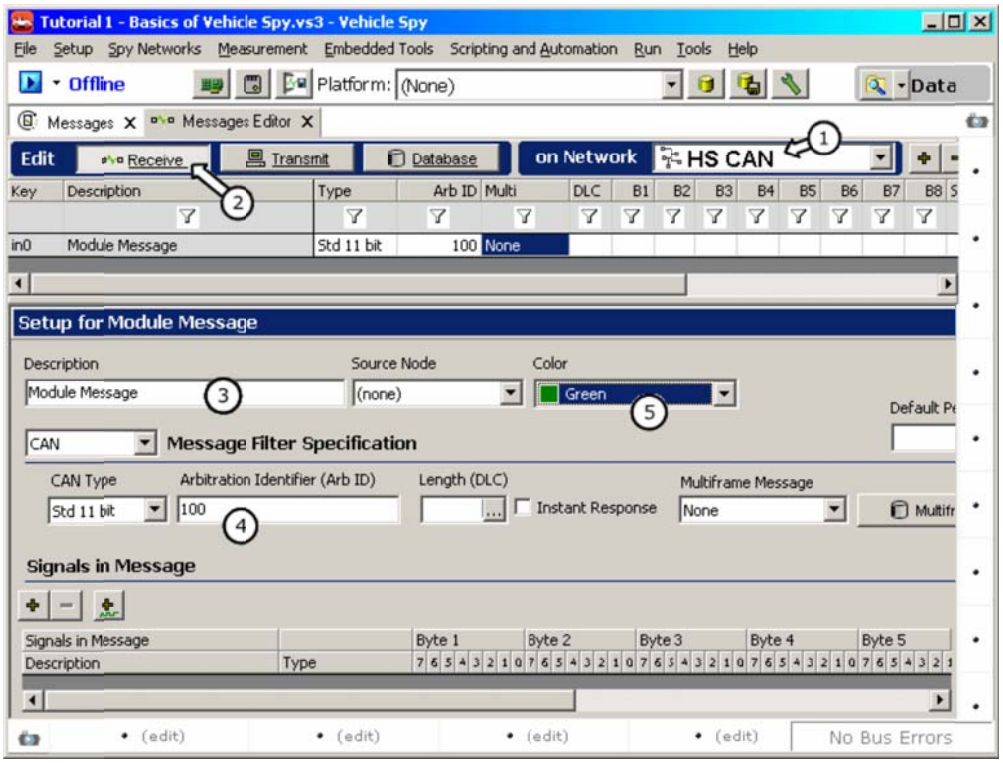
メッセージを固定表示へと切り替えるには、再度 Scroll ボタンを押します。少なくとも 1 回は検出された、すべてのメッセージが表示されます。直近[3]で変化のあったメッセージは、灰色で強調されます。ある程度の長時間経過後[4]に現れるメッセージは、最初、青色で強調され、徐々に白へと変化してゆきます。

4) **Vehicle Spy 停止：**Messages ビューの表示に慣れてきたら、再度スタートボタンを押して Vehicle Spy を停止します（①）。

パート３：メッセージの詳細を確認

メッセージトラフィック中の、ただ 1 つの特定のメッセージを観測したいとします。Messages ビューは、観測対象を的確に表示するよう設定でき、完全にカスタマイズ可能です。次の数ステップで、Vehicle Spy のフィルタ機能を体験するのに適した、基本的なメッセージの仕様を設定してみましょう。

1) **Messages Editor ビューに切り替え：**メニューバーから Vehicle Networks を選択し、Message Editor を選択します。メインワークスペースのパネルが Message ビューから Messages Editor に変わります。



2) **新しいメッセージ仕様を作成：**このチュートリアルでは、仮に受信したある High Speed CAN メッセージを特定したいとします。Network ドロップダウンから HS CAN が選択され（①）、Receive ボタンが

[3] 訳注：１秒以内の変化(Vehicle Spy 3.5.3.56 における仕様)
[4] 訳注：１秒以上の変化(Vehicle Spy 3.5.3.56 における仕様)

押されていることを確認して下さい（②）。新規のメッセージ仕様を作成するには、パネル上部の”＋”ボタンをクリックします(メッセージ仕様は、簡単に削除可能です。Vehicle Spy 停止時に、削除したい行をクリックし、さらに”－”ボタンをクリック）。空の行がパネル上側の表部分に、Setup ダイアログが下側に表示されます。この Setup ダイアログで、新規の仕様を定義します。

3) **Description 欄に Module Message を入力：**次に、メッセージに記述を加えます。これを Module Message と呼びます。Setup ダイアログの Description 欄をクリックし、”Module Message”と入力してみましょう（③）。

4) **Arbitration ID[5] “110”を入力：**ここで、“110”を Arb ID 欄に入力します（④）。これにより、”Module Message”という記述が、ID 110 のメッセージに対して付加されます。

5) **メッセージに色をつける：**注目するメッセージに色をつけると便利です。大量のメッセージが流れている中で、独自の色をつけることで、瞬時にそのメッセージの認識が可能になります。Color 欄の右の小さな矢印をクリックし（⑤）、ドロップダウンから色を選択してください。

6) **完成したメッセージの確認：**このチュートリアルで指定すべき項目は、以上です（Setup ダイアログ中の各オプションの詳細については,5.3.3 章を参照して下さい）。Messages Editor パネルの表部分で、これまでの入力内容を確認して下さい。Setup ダイアログに入力したすべての仕様は、自動的にこの表に反映されます。入力したメッセージが、上図と全く同じとなっていることを確認して下さい。

   備考:　好みに応じて、メッセージ記述や Arbitration ID のような情報は、直接表中に、もしくは Setup ダイアログに入れることができます。一方での変更は、もう一方にも自動的に反映されます。

パート４：実行中にメッセージの詳細をチェック

以上で、すべての設定が完了しました。Messages ビューに切り替えて、これまでの作業内容を見てみましょう。

1) **Messages ビューを開く：**メインワークスペースの上部のタブから、Messages ビューに戻って下さい。

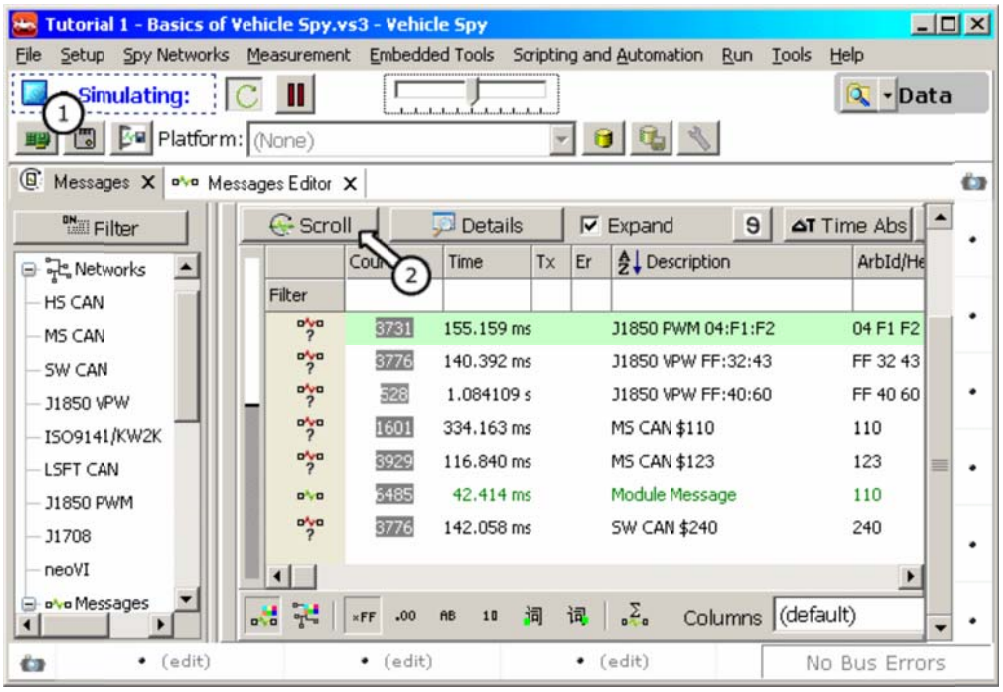
2) **Vehicle Spy の開始：画面**左上のスタートボタンをクリックします（①）。

3) **スクロール・モードに変更：**Scroll ボタンを押します（②）。Arb ID 110 の HS CAN メッセージに紐付けられた、”Module Message”という記述が現れることを確認して下さい。Pause ボタンを押せば、より簡単に確認できます。

4) **固定モードに変更：**ここで、Scroll をオフにします。先に指定した色の着いた、”Module Message”が見えます。メッセージ中のデータの変化はいかがですか？データの変化は、灰色で強調されるため容易に見つけることが可能です。

   備考：編集作業中に、Vehicle Spy を停止(画面左上のスタートボタン押下)する必要はありません。

[5] 訳注：アービトレーション ID。メッセージを識別し、その優先順位を示す ID

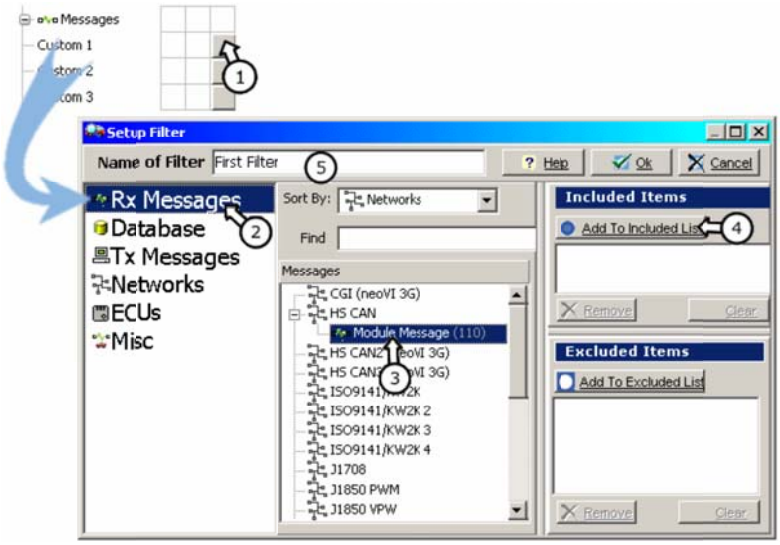


パート５：”Module Message”のみを表示(フィルタリング)

完成したメッセージ仕様は、カスタムフィルタ[6]を作成する上での基本要素となります。本パートでは、自分独自のカスタムフィルタを作成してみましょう。

1) **Messages ビューを開く：**Messages タブを選択します。パネルが開くと、左側に Filter バーが自動的に表示されます。これは任意のフィルタをオン/オフする際に有用です。リスト中の Custom 1 の、右側の小さなボックスをクリックし（①）、フィルタを定義する Setup Filter ダイアログを開いて下さい。

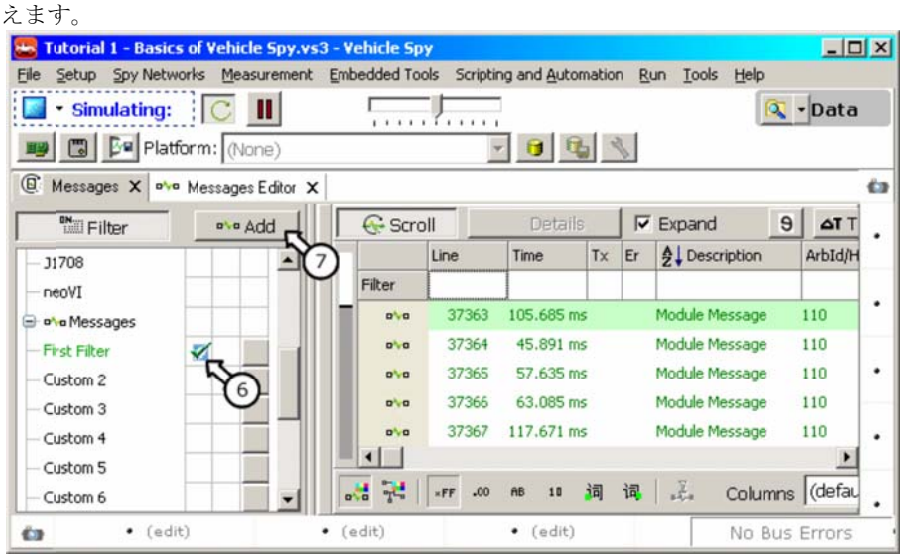
[6] 訳注：Messages ビュー上に設置されている既定のフィルタ(パート６参照)に対する、カスタマイズ可能なフィルタ



2) **カスタムフィルタを作成：**Setup Filter ダイアログでは、フィルタを簡単に設定可能です。ダイアログの左側にメッセージ源のリストがあり、ここから、Messages Editor で定義したメッセージにアクセスが可能です。Rx Messages オプションが選択されていることを確認して下さい（②）。High Speed CAN の下のリストに自身で作成した Module Message が現れます（③）。Vehicle Spy は、作業効率を極限まで高めるために、定義した全メッセージをしかるべき場所に自動的に配置します。Module Message をクリックし、Add to Included List ボタンを押下して下さい（④）。これで、Module Message がフィルタに追加されます。

3) **フィルタの命名：**複数のフィルタを使用する場合、名前を指定しておくと、管理上有用です。Name of Filter 欄をダブルクリックし、First Filter と名前を付けてみて下さい（⑤）。完了後、OK ボタンを押し、Messages ビューに戻って下さい。

4) **フィルタを有効化：**ここで、Filter バーを注意深くみて下さい。Custom 1 の表題 が、First Filter に置き換わりました。画面左上のスタートボタンを押下して Vehicle Spy を開始し、フィルタ名の右側最初の欄をクリックします（⑥）。するとその欄に、緑のチェックマークが表れ、First Filter の文字列が緑に変わり、メッセージトラフィックが著しく変化します。バス上の全メッセージではなく、Module Message だけが見えます。



5) **カスタムフィルタを追加：**カスタムフィルタを追加するには、Filter バーの上の Add ボタンを押します（⑦）。追加したフィルタを削除する場合、フィルタ上で右クリックし Remove Message Filter を選択して下さい。

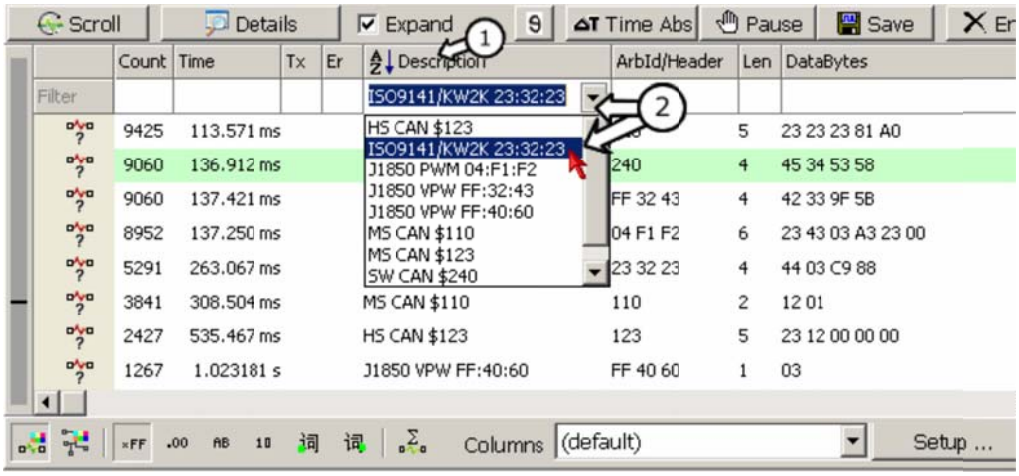
フィルタリングに関する備考：Vehicle Spy のフィルタは、メッセージの包含(インクルード)と同様にメッセージの除外（エクスクルード）が可能です。これが正しく動作するためには、除外する前にあらかじめ、何らかのメッセージを包含しておく必要があります。Vehicle Spy は、包含したメッセージを最初に探し、そしてその包含リストから、除外を行います。例えば、メッセージトラフィックから Module Message を除外する場合を考えます。はじめに、Filter パネルの、Data Types 下の Network の右隣の列をクリックし、緑のチェックマークを置きます。Network フィルタは、単純に、全ネットワークトラフィックを包含します。次に、First Filter の右側 2 番目の欄をクリックします。その欄に X がつき、First Filter の文字列が赤になり、Module Message はメッセージトラフィックから除外されます。

Filter パネルについて、もう少しだけ、色々と試してみましょう。全ネットワークトラフィックを包含し、かつ全 HS CAN メッセージを除外します。また、全 HS CAN メッセージを含め、Module Message を除外してみて下さい。Filter パネルによって、データ表示をカスタマイズする方法が無限に広がります。

パート６：その他のフィルタ

カスタムフィルタ以外にも、Vehicle Spy は様々なフィルタリング方法を提供します。

1) **メッセージ列の整列：**Vehicle Spy を、シミュレーションモードでオンライン[7]にして下さい。Messages ビューの列のタイトルに注目します。Description をクリックすると、アルファベット順に整列する記号が文字列の隣に現れ、メッセージ列は A から Z の順に整列されます。同じ場所を再度クリックすると、Z から A の順になります。大量のメッセージを観測する場合、この方法によって、スクロールバーに頼ることなく容易に前後移動が可能です。他のタイトルをクリックすると、新しい列の整列へと焦点が移ります。



これら列フィルタによって、カスタムフィルタを設定することなく、大量のメッセージに対する素早く効率的な整列が可能です。

[7] 訳注：オンライン = Vehicle Spy の画面左上のスタートボタンが押された状態（即ち Vehicle Spy と車載ネットワークが接続[オンライン]状態）

2) **ドロップダウンリストを使用した列フィルタ：**各列のタイトルの直下には、列フィルタのために確保された空欄があります。Description の下の空欄をダブルクリックしてみましょう。ドロップダウンリストを示す右側の小さな矢印をクリックします。ドロップダウンリスト内に、Messages ビューに現れる全てのメッセージ記述が見えます。そのうちの 1 つをクリックし、Enter キーを押して下さい。空欄がメッセージ記述で埋まり、その記述を持つメッセージのみが表示されます。このフィルタリング方法は、Arbitration ID のような、他の列に関しても同様に適用されます。

3) **入力欄を使用した列フィルタ：**大量のメッセージを観測しているとしましょう。探している特定のメッセージは、その Arbitration ID に 6 を含みますが、正確な値がわかりません。そのような場合、Arb ID の列のタイトル直下の空欄に 6 を入力し、Enter キーを押します。640, 126, 260 のような 6 を含む Arbitration ID を持つメッセージのみが表示されます。

   他の例として、探している ID が 3 で終わるとします。??3 と入力すると、Vehicle Spy は 123, 333, 643 のような 3 で終わるすべての Arbitration ID を示します。フィルタに 1 つ以上のメッセージを含めたい場合は、“?2?, 110, ?33” のようにコンマで区切ります。これらの列フィルタは、他の列に対しても同様に動作します。もしデータベースを使用し、すべてのメッセージとシグナルが定義されている場合、Description 列の直下の空欄に eng または odo と入力してみて下さい。メッセージ記述にそれらの文字を含むメッセージやシグナル、即ちエンジンとオドメータに関係するものだけが残ります。

   列フィルタを削除するには、該当する場所をクリックして強調表示し、Delete キーを押して下さい。

   フィルタを変更することなく全ての列フィルタを有効化・無効化する場合は、Messages ビュー左上の Filter ボタンを使用します。

   備考:　列の整列およびフィルタは、オンライン時でもオフライン時でも編集可能ですが、その変更は、Vehicle Spy がオンラインのときのみ反映されます。

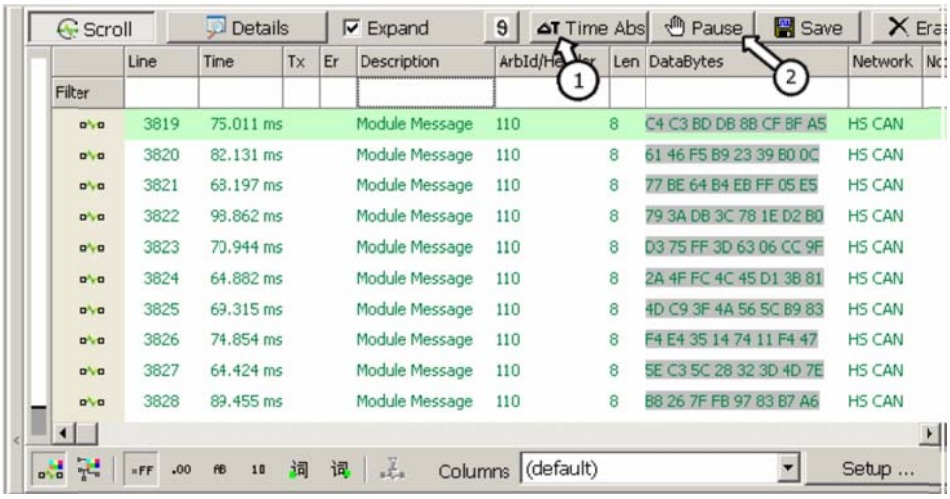
パート 7 ：メッセージのタイムスタンプ

Filter パネルの試行が終了したら、カスタムフィルタによって Module Messages を包含した場面にまで戻って下さい。本パートでは、Messages ビューのいくつかの重要な機能を観てみましょう。

1) **相対・絶対時間表示：** Messages ビューの Scroll ボタンをクリックして、スクロール表示にして下さい。画面中の Time 列に気づかれたかもしれません。各メッセージは、それぞれ固有のタイムスタンプと共に記録されます。デフォルトでは、Messages ビューは各メッセージに対して、相対時間表示を行います。相対時間とは、現在のメッセージと、その直前のメッセージとの時間差です。絶対時間表示に切り替えるには、Messages ビューの上部の ΔT Time Abs ボタンをクリックします.（①）。絶対時間とは、Vehicle Spy 開始時[8]から数えて、そのメッセージが現れたときの時間です。

2) **一時停止：**Messages ビュー使用時、一時停止機能は非常に重要です。スクロール表示をオンにして、ΔT Time Abs ボタンの右側の Pause ボタンをクリックして下さい（②）。スクロールが停止し、パネル右側にスクロールバーが表示されます。過去のメッセージトラフィックを観るためには、スクロールバーをクリック＆ドラッグ、もしくは Page Up/Down キーを使用します。Pause ボタンは、スクロール表示、固定表示時の両方で同等に機能します。

[8] 訳注：スタートボタン押下（オンライン移行）時。Vehicle Spy の起動時ではありません。



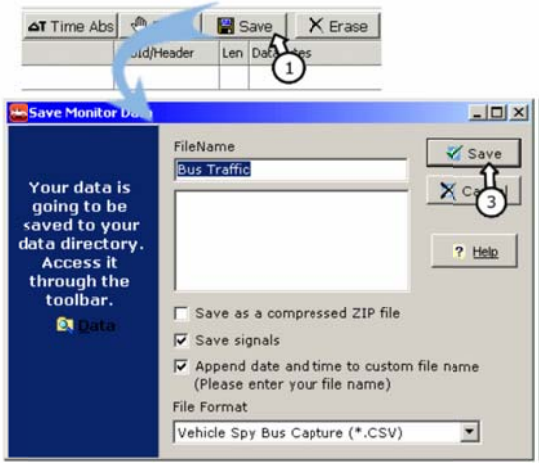
パート８：メッセージビューバッファ[9]の保存

後で別途解析するため、収集したデータを保存しておきたい場合があります。

1) **収集したバッファをハードディスクに保存：**Messages ビュー上部の Save ボタンをクリックします（①）。

メインワークスペース上に Save Monitor Data ダイアログが開きます。自分で入力したデータファイル名、もしくは時刻と日付を含むデフォルトのデータファイル名を使用することができます。

File Name の空欄の下に、さらに追加の空欄があります。これは、データとともに注釈を入力したい場合に使用します[10]。完了後、Save をクリックし、データを保存します（③）。



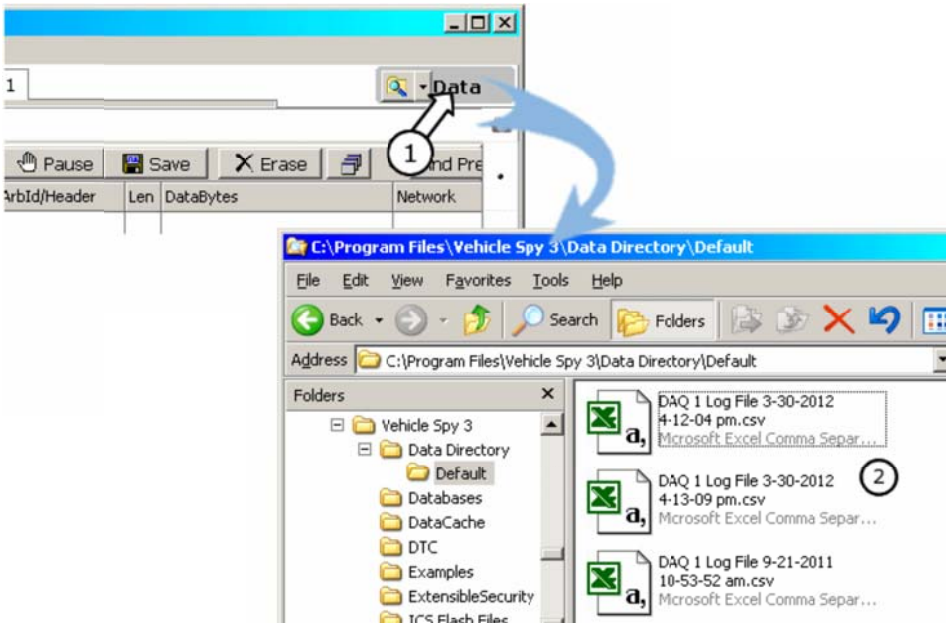
[9] 訳注：バッファ＝観測したメッセージ群を**一時的に**保存しておく PC 上のメモリ領域（デフォルトで 50,000 メッセージ）。観測を続けると、古いデータから順に、新しいデータで書き換えられてゆきます。

[10] 訳注：CSV ファイル中の Note という項目に、ここでの入力内容が反映されます。

2) **作成したデータファイルを開く：**手順 1 で OK をクリック後、データが自動的に Data Directory に保存されます。これは基本的に、Logon 名と同一の名前を持つサブディレクトリです。これは、全てのデータと各種設定に対する、デフォルトの保存場所として扱われます。保存したファイルに素早くアクセスするには、メニューバー右側の Data ボタンをクリックします（①）。

作成したデータファイルは、データフォルダ中にあります。データファイルをダブルクリックして下さい（②）。

そのファイルは CSV ファイル用の、デフォルトのエディタで開かれます。もし Microsoft Excel がインストールされていれば、通常、それがデフォルトのエディタです。Excel がなければ、メモ帳になるでしょう。Vehicle Spy でこれらのファイルを表示および解析用に開くこともできます。



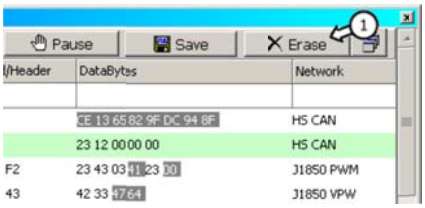
パート９：メッセージビューバッファの消去

時には、Messages ビュー上で収集した全データを消去したい場合があります。

1) **Messages ビューを選択：**Messages ビューのタブを選択します。

2) **バッファを消去：**Messages ビュー右上の Erase ボタンを押します（①）。

3) **Messages ビューの消去を確認：**メッセージカウンタ列が再び 1 から始まることを確認して下さい。



例題

以上で、Vehicle Spy の全ての基本機能を網羅しました。確認までに、以下にこれまでの内容をまとめます。

- Logon ダイアログにて設定ファイルをロード
- スクロール表示と固定表示の両方で、メッセージトラフィックを観測
- Messages Editor ビューにおけるメッセージの設定
- Messages ビューでメッセージを表示
- カスタムフィルタを設定
- 列フィルタを使用したフィルタリング
- フィルタの有効化/無効化と、包含/除外の切替
- 相対/絶対タイムスタンプ
- データの一時停止
- 収集したデータのハードディスクへの保存
- Messages バッファを消去

より複雑な課題に取り組む前に、次の例題にトライしてみましょう。これまで学んだ内容の強化に役立つことでしょう。

例題

1) Filter バーを用いて、下記のフィルタの組み合わせをそれぞれ設定して下さい。また、それがメッセージトラフィックに与える影響を説明して下さい。

   a. HS CAN を包含
   b. HS CAN と J1850 VPW を包含
   c. Network を包含、HS CAN を除外（Network は Filter パネル内の、Data Types の下に位置）

   ※包含＝✔マーク、除外＝✕マーク

2) 以下の両方を行って下さい。

   a. Messages Editor にて、あるメッセージ仕様を設定して下さい。それは、J1850 VPW メッセージで、１６進数のデータバイト”FF 40 60 03”を含みます。
      “FF” = Priority Type
      “40” = Target
      “60” = Source
      “03” = Byte 1

   b. このメッセージを包含するカスタムフィルタを作成して下さい。このメッセージのみを、Messages ビューに表示してください。

3) Arb ID “123”の CAN メッセージと、ヘッダ”FF”の J1850 メッセージ全てをフィルタする列フィルタを作って下さい。ヒント：コンマの使用

いかがでしたか？
少し休憩を挟んでから、次のチュートリアルに進みましょう。

チュートリアル２：メッセージ送信

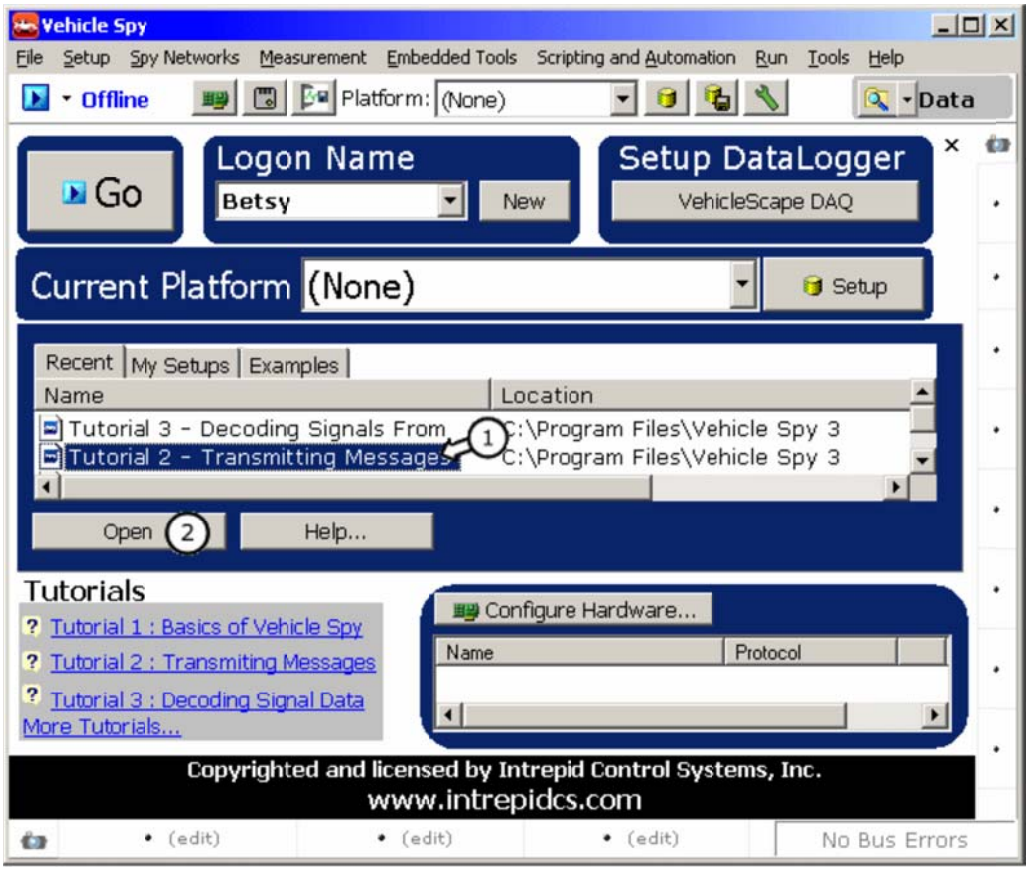
本チュートリアルでは、メッセージを送信する各種方法を学びます。また、重要なユーザインタフェース機能、代替ワークスペースについても触れます。

パート１：セットアップファイルのロード

本チュートリアルでは、Vehicle Spy による、車載ネットワーク上でのメッセージ送信方法を学びます。

1) **Logon 画面に戻る：**これには 2 種類の方法があります。１つ目はすでに Vehicle Spy を起動している場合で、オフライン状態であることを確認してからメニューバー左上の File をクリックします。そしてドロップダウンから Logon を選択します。２つ目は Vehicle Spy を起動していない場合で、デスクトップの Vehicle Spy アイコンをダブルクリックすれば Logon 画面が自動的に開きます。

2) **Tutorial 2 を開く：**Logon パネルの Examples タブをクリックします。Tutorial 2 – Transmitting Messages を、スクロールして見つけて下さい（①）。それをクリックし、Open ボタンを押下して下さい（②）。



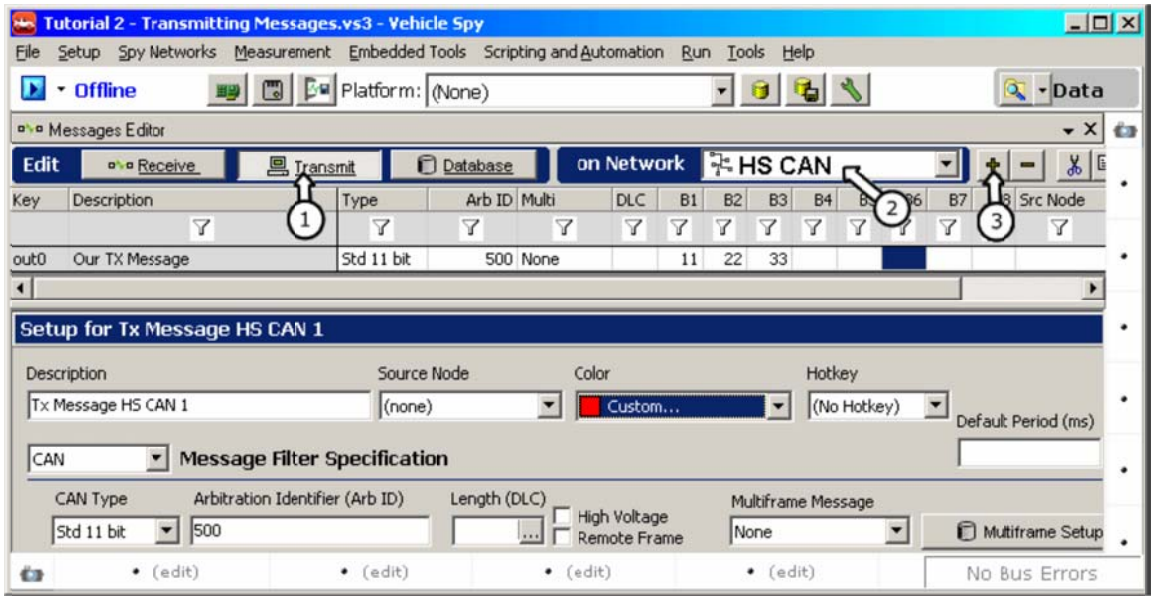
パート２：送信メッセージの入力

チュートリアル 1 では、Messages Editor ビューで受信メッセージの仕様を設定しました。ここでは改めてこのビューに戻り、送信メッセージの設定を行います。

1) **Messages Editor パネルを開く：**メインワークスペース上部のタブから Messages Editor ビューを開きます。このビューは、Spy Networks -> Messages Editor でも開くことができます。

2) **送信メッセージ表に切り替え：**Message Editor パネルの上部の Transmit ボタンをクリックします（①）。すると、送信メッセージの設定画面に切り替わります。次に、on Network ドロップダウンから HS CAN を選択します（②）。ここで、新しいメッセージを追加するために＋ボタンをクリックします（③）。

3) **送信メッセージを定義：**＋ボタンをクリックすると、Setup ダイアログが開きます。ここで、Description、Arb ID、Color 等の送信メッセージの仕様を指定します。HS CAN にて Our TX Message という名のメッセージを作成してみましょう。Arb ID を 500、データバイトを 11-22-33 にします（表の B1 列から順に入力）。他の仕様に関しては、後で扱います。



4) **完成したメッセージの確認：**Setup ダイアログに追加した内容が、自動的に上の表に反映されることを確認して下さい。

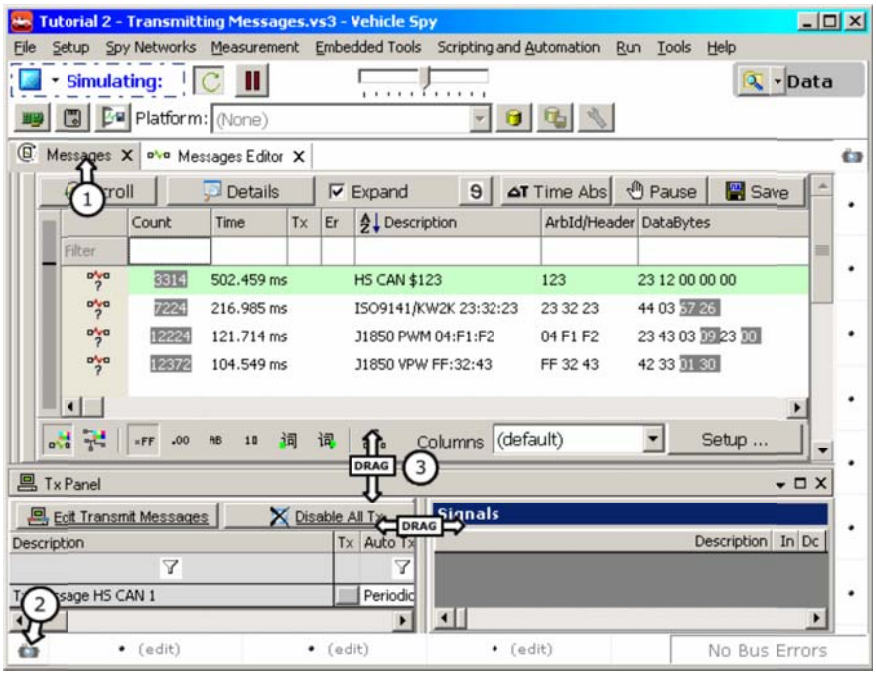
パート３： 代替ワークスペースの活用

Vehicle Spy ユーザインタフェースの紹介時に、ユーザインターフェースの下側と右側に格納された代替ワークスペースについて触れました。本パートでは、Tx Panel と Messages ビューの両方を同時に表示するために、このパネルを 1 つ使用します。

1) **Messages ビューを開く ：**Spy Networks メニューから Messages を選択します（①）。これまでどおり、メインワークスペースに Messages ビューが現れます。

2) **代替ワークスペース(下側)を設定：**ユーザインタフェースの左下の View アイコンをクリックします（②）。Spy Networks を選択し、ドロップダウンから Transmit Panel を選択します。すると、Messages ビューはそのままで、ユーザインタフェースの下側の代替ワークスペース中に、Tx Panel が開きます。

代替ワークスペースのサイズを変更するには、2 つのウィンドウ枠の間をクリックし、希望のサイズにドラッグします（③）。



現状では、この機能はあまり重要でないように見えるかもしれません。しかし次のパートでは、その有用性がよくわかることでしょう。

パート４：メッセージの手動送信

今、Tx Panel と Messages ビューが開いているはずです。Tx Panel をよく見て下さい。Our Tx Messages がすでにあります！Vehicle Spy では、 余計なタイピングやコピー/ペーストが省けるので、作業効率が高まります。では、Tx Panel の動作を見てみましょう。

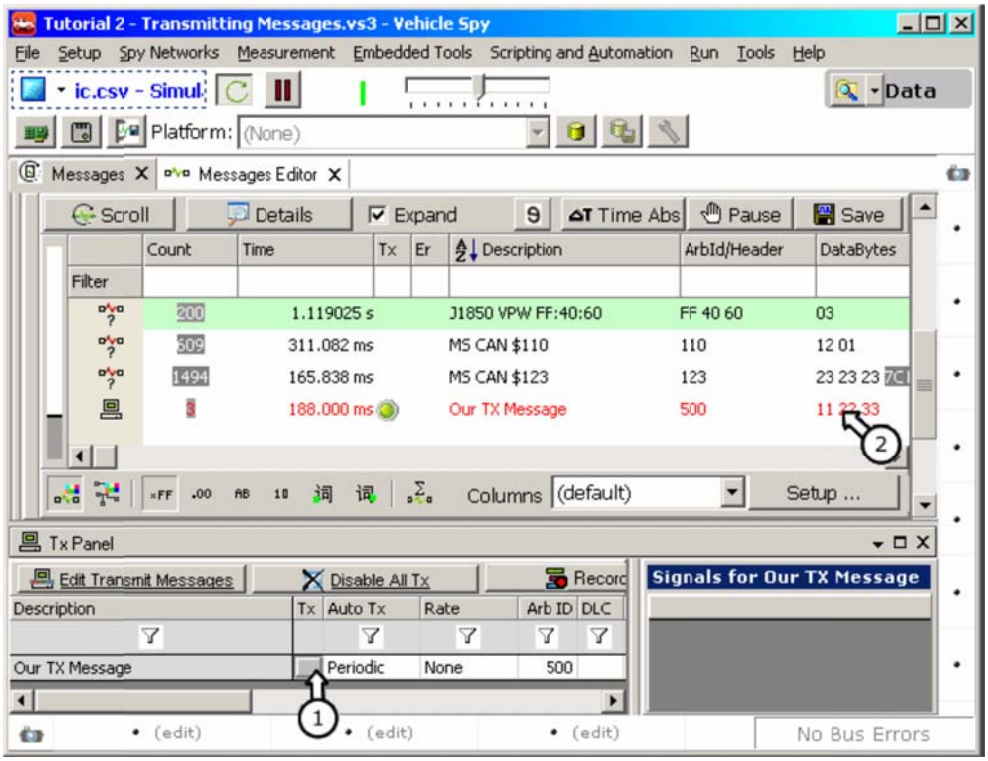
1) **Vehicle Spy の開始 ：**Messages ビューでデータの流れが見えることを確認します。

2) **手動送信ボタンをクリック：**Tx Panel の Our TX Message の右側に灰色の Tx ボックスがあります（①）。このボックスをクリックし、メッセージを送信してみて下さい。

   Messages ビューが固定表示の場合、Our TX Massage は、メッセージ中に表れるはずです。最近の活動を示すために、初めは強調表示されます。再度、直ちにメッセージを送らなければ、強調表示は消えてゆきます（②）。

   スクロール表示の場合、何も見えないかもしれません。その場合、送信して Messages ビューを見るまでに、メッセージが流れ去ってしまったのでしょう。Pause ボタンをクリックしてスクロールすることで、そのメッセージを見つけることが可能です（Tx Messages は、簡単に識別できるように緑の LED ア

イコンでマークされます）。また、今度はより細心の注意を払って、再度、Tx ボタンをクリックしてみましょう。過ぎ去ってゆくメッセージを捉えられるはずです。すべての Tx Messages には、Tx 列に緑の LED 記号が付きます。これは、そのメッセージが送信メッセージであることを強調します。

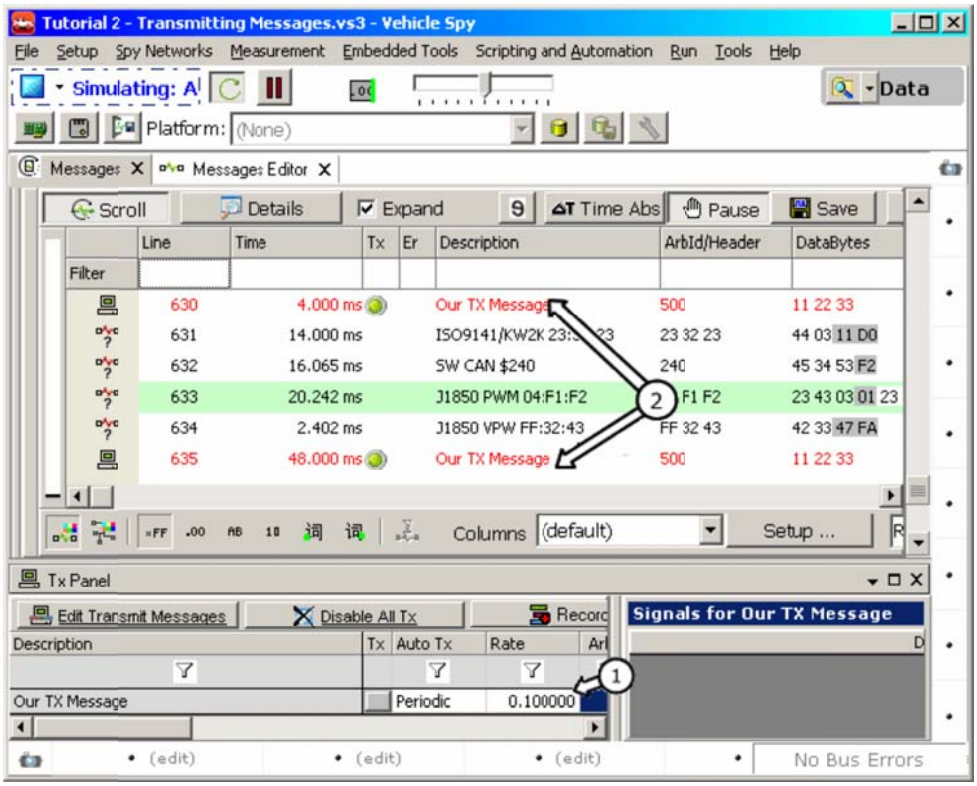


パート５：メッセージの周期的送信

しばしば、メッセージを一定間隔で送信する必要があります。Tx ボタンを一日中押し続ける必要はなく、Vehicle Spy では、メッセージをいつ、どれ程の頻度で送信するかを指定することができます。

1) **Auto Tx と Rate の設定：**改めて Tx Panel を見、Auto Tx と Rate 列に注目して下さい。以前、これらには触れませんでしたが、今回はその設定に入ります。Auto Tx コラムの変更は、メッセージの送信方法に影響します。本チュートリアルでは、Periodic のままにしておきます。次にその右側、Rate 列の空欄をダブルクリックして下さい（①）。現れたドロップダウンの矢印をクリックすると、一般によく用いられる値のリストが表示されます。そこに無い値が必要な場合、その空欄に直接入力することも可能です。本チュートリアルでは、ドロップダウンから 0.1 を選択して下さい。以上で、Vehicle Spy は Our TX Message を 0.1 秒毎に送信するようになりました。

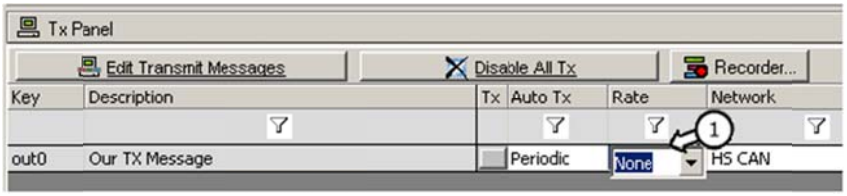
2) **メッセージトラフィックの確認：**以上の設定完了の後、Messages ビューを見て下さい。スクロール表示の場合、Tx メッセージが一定間隔で流れるのが見えるはずです（②）。固定表示の場合、Our TX Message が強調され、メッセージカウンタが急激に増加してゆくでしょう。



パート６：メッセージへの自動応答

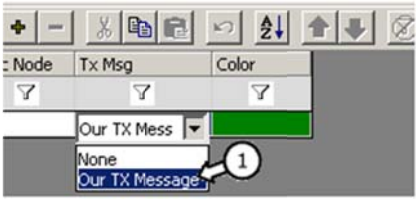
手動送信と周期的送信に加え、Vehicle Spy はメッセージに対する自動応答が可能です。

1) **送信メッセージを手動モードに戻す：**周期的送信モードを解除するため、Our TX Message の Rate 列を None に設定します（①）。

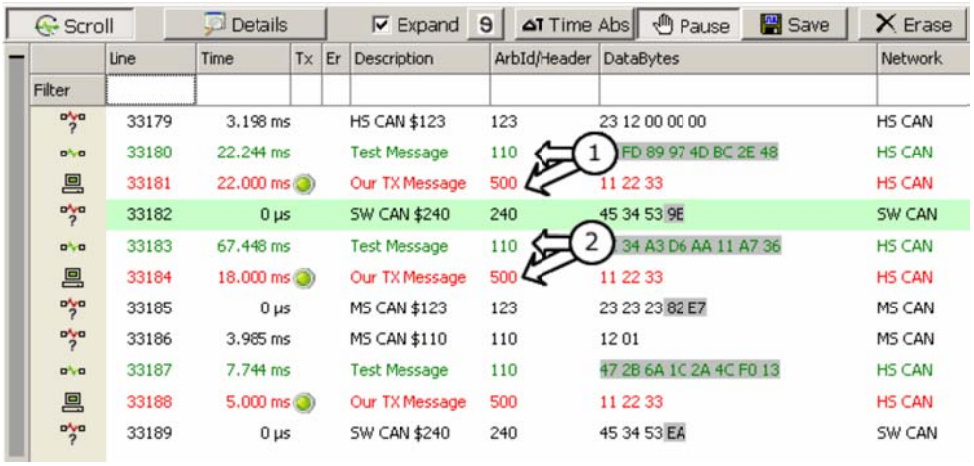


2) **Messages Editor の Receive 表に移動：**メインワークスペースのタブから、Messages Editor を開きます。Test Message という名のメッセージが、このチュートリアルのため、すでに Receive 表に設定されている点に注意して下さい。

3) **自動応答を設定：**表中にて、Test Message に対応する Tx Msg 列をダブルクリックします（①）。定義済みの送信メッセージを含むドロップダウンが開きます（この場合は Our TX Message のみですが、複数の送信メッセージを定義した場合、それらも全てここに表示されます）。Our TX Message を選択し、Enter キーを押します。以上により、Vehicle Spy は Test Message を受信するたびに Our TX Message を送信するようになりました。



4) **メッセージトラフィックを確認：**Test Message が現れるごとに、Vehicle Spy が Our TX Message を送信し、応答していることが分かります（スクロール表示にて見やすいですが、Pause が必要かもかもしれません）。Messages ビューは下図のように見えるはずです（①&②）。



5) 捕捉：さらなるトレーニングとして、**Test Message** と **Our Tx Message** 両方を包含するカスタムフィルタの作成、また列フィルタの作成にトライしてみて下さい。

例題

本チュートリアルを通じて、以下を習得しました。

- 送信メッセージを作成
- 代替ワークスペースを利用
- メッセージの手動送信
- メッセージの周期的送信
- 受信したメッセージに対する自動応答

Vehicle Spy のエキスパートに、順調に近づきつつあります！

チュートリアル３に進む前に、以下の例題を試してみて下さい。

例題：

1) a) ２秒毎に送信するメッセージを作成して下さい。
   b) 送信メッセージのみを表示するフィルタを作成して下さい。
      ヒント：Filter パネルの Data Type 以下を参照して下さい。

2) a) Arb ID 240(16 進数)の SW CAN メッセージを作成して下さい。
   b) データバイト FF 22 34 12 の J1850 VPW 送信メッセージを作成して下さい。
      そして、SW CAN メッセージを受信するごとに、このメッセージを送信するよう設定して下さい。

以上です。準備ができたら次のチュートリアルに進んで下さい。

チュートリアル３：メッセージからシグナルをデコード

概要

多くのネットワークメッセージには、シグナルと呼ばれる情報が含まれています。シグナルの種類としては、アナログとデジタルが挙げられます。デジタル情報は、Yes/No，On/Off，Open/Close 等が考えられます。ライトの On/Off とドアの Open/Close は、この種類に該当します。一方、アナログ情報は、２通りの単純な状態に限定されず、可変です。それは例えば、車速、エンジン回転数、冷却温度、またはその他の変動しうる値が挙げられます。

チュートリアル 3 では、Vehicle Spy でアナログ情報を抽出または解読します。デジタル信号の解読も、基本的に同じ方法で行われるので、本チュートリアルの内容が有効です。

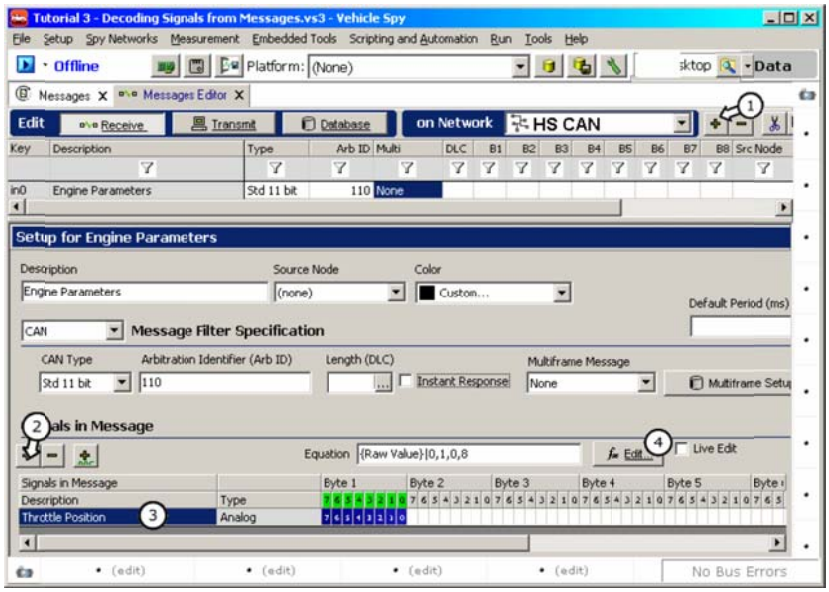
パート１：設定ファイルのロード

1) **チュートリアルファイルのロード：**これまでのチュートリアルのように、オフライン状態の Vehicle Spy で Logon 画面に戻り、Examples タブをクリックし、ファイル "Tutorial 3 – Decoding Signals from Messages" を選択します。

パート２：メッセージ入力

1) **メッセージの作成：**タブを使用して、メインワークスペース上に Messages Editor パネルを開きます。HS CAN を選択し、＋ボタンをクリックして、HS CAN の新規の受信メッセージを定義します（下図①）。

Setup ダイアログから、以下の仕様を入力します。
Description:　Engine Parameters
Arb ID:　110



2) **シグナルの作成：**Setup ダイアログの下部に、Signals in Message エリアがあります。このエリアの＋ボタンをクリックして、空のシグナル仕様を追加して下さい（②）。

3) **シグナル記述の入力：**そのシグナルに、Throttle position という記述を設定して下さい。これは、デフォルトの記述（③）をダブルクリックし、新しい記述をタイピングすることで行います。

4) **シグナル・デコードの設定：**デコードオプションを表示するために Edit ボタン（④）をクリックし、このシグナルのデコード方法を定めます。

パート３：アナログ値に対して式を入力

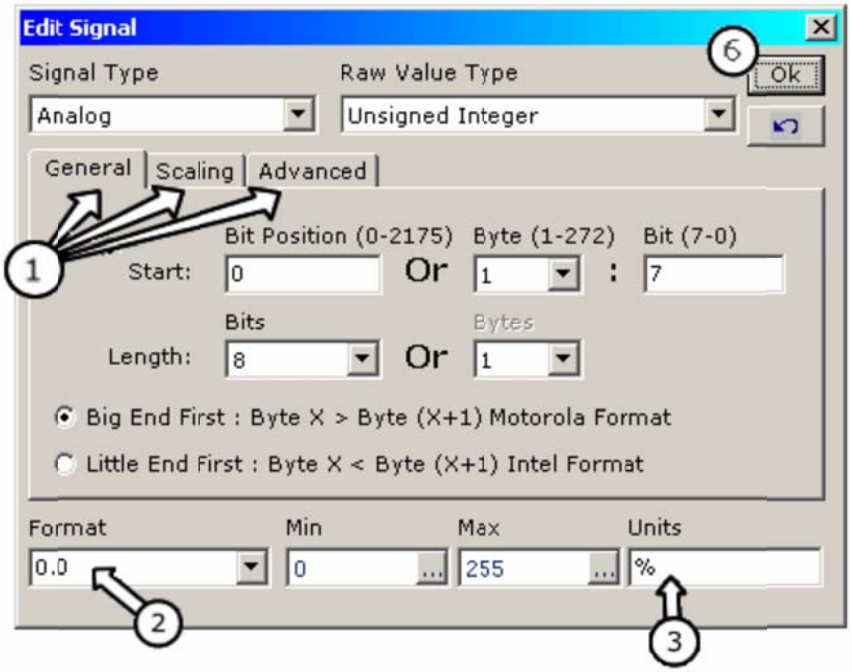
Throttle Position シグナル値のデコードにあたり、適当な式を入力する必要があります。

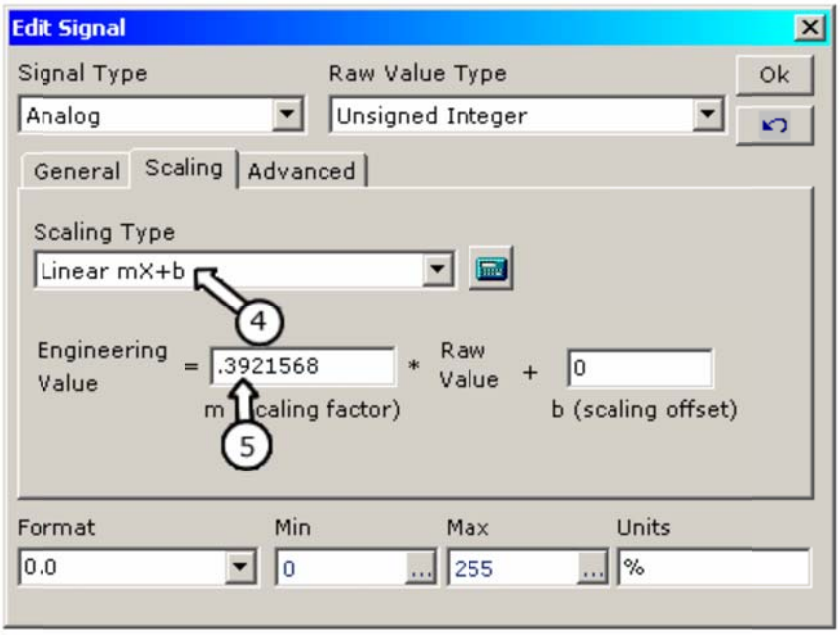
1) **Signal Type の選択：**Throttle Position は、多様な値を取り得ます。可変であり、即ちデジタルではなくアナログと言えます。Signal Type は、Edit Signal ダイアログのドロップダウンリストから選択し、この例ではアナログを選択します。

2) **シグナル位置の選択：**次に、メッセージ中のどのビットとバイトをデコードしたいのか、その位置を指定する必要があります。Start 欄の、Bit Position が 0 に設定されていることを確認して下さい。Length は 8 ビットである必要があります。これにより、Engine Parameters メッセージの最初の 1 バイトが、Throttle Position の生データとして設定されます。エンディアンは、デフォルトの Big End First のままにして下さい。

3) **シグナルフォーマットの選択：**Format 欄は、デコードされたデータの表示方法を指定します（下図②）。リストから 0.0 を選択して下さい。すると、シグナルは小数点以下 1 桁の数値で表示されます。

4) **単位の入力：**Units 欄に % を入力します（③）。これにより、スロットル位置が%単位で表示されます。





5) **式の編集：**Scaling タブを選択します（①）。ここで、生データから、実際の工学値に変換するための式の設定を行います。Linear mX+b を選択して下さい（④）。Engineering Value 欄に 0.3921568 のスケーリング定数を入力します（⑤）。この値は、8 ビット値を%にスケーリングします。（0.3921568 = 100 % / 255 bits）

6) **Raw Value Type の選択：**デフォルトで、Raw Value Type は Unsigned Integer に設定されます。

7) **式の完了：**Ok を押して、編集を完了して下さい（⑥）。

備考: デジタルシグナルをデコードする場合、手順はほぼ同じです。単に Signal type で Digital を選択し、シグナルのビット位置を入力します。

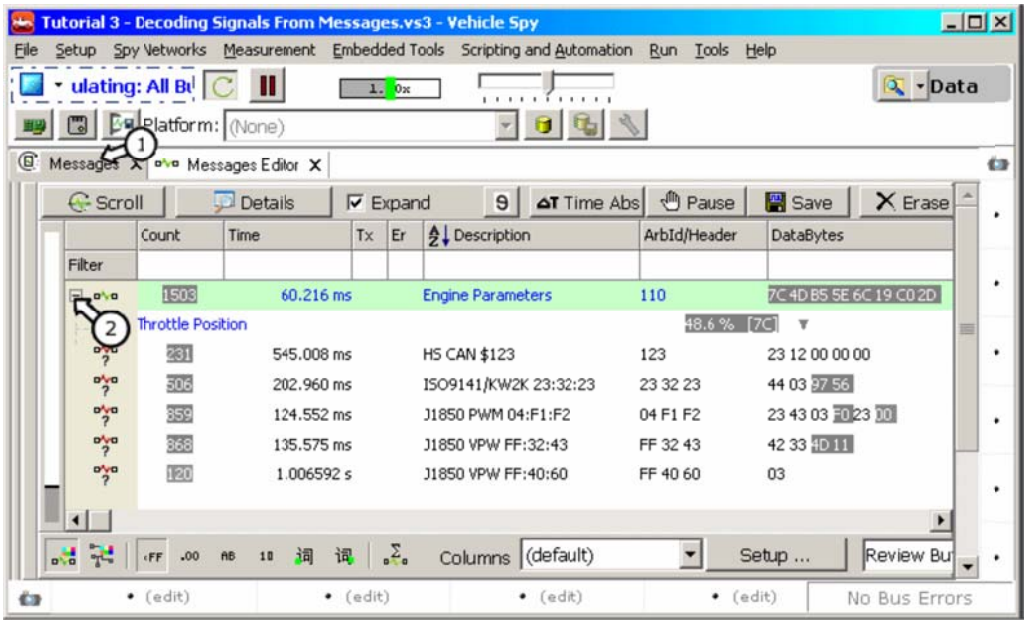
パート４：メッセージデータ表示

ここで、作成したアナログシグナルの動きを見てみましょう。

1) **メッセージフィルタの作成：**これまでのチュートリアルの内容を活かして、Engine Parameters メッセージだけを表示するカスタムフィルタを設定します。本チュートリアルでは特に重要ではありませんが、大量のメッセージを扱う場合、このようなフィルタが非常に役に立ちます。

2) **メッセージトラフィックの表示：**固定表示で、Messages ビューに切り替えます。指定した色のついた Engine Parameters メッセージに注目して下さい。

3) **メッセージを展開してシグナルを表示：**メッセージ記述の左側に、小さな＋があります（②）。これをクリックして、メッセージを展開して下さい。すると、Engine Parameters メッセージ内に定義した Throttle Position シグナルが見えます。もし複数のシグナルを定義していれば、それらも同様にここに表示されます。もう一度メッセージだけを見るには、－をクリックして下さい。



4) **スクロール表示に切り替え：**Messages ビューをスクロール表示に切り替えます。ここで、フィルタをオンにします。フィルタを有効にすると、Engine Parameters メッセージだけが見えるはずです。Pause を押して下さい。固定表示時と同様、メッセージ記述の左側に＋が現れる点に注意して下さい。メッセージ内で定義されたシグナルを見るために、再度＋をクリックしてみましょう。

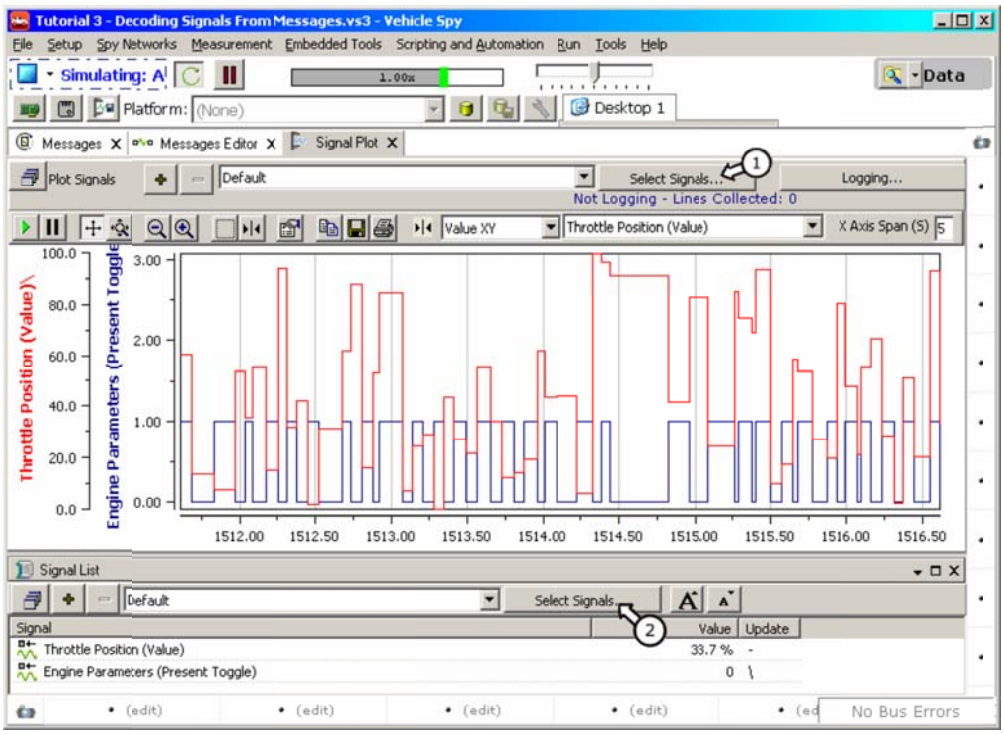
備考：Activity Highlighting は、メッセージと同様、シグナルにも自動的に適用されます。強調表示に関してよく覚えていない場合は、チュートリアル 1 のパート 2 を参照して下さい。ここで再度、Activity Highlighting の基本を説明します。

- 固定表示では、少なくとも１回、バスで検出された全てのメッセージとシグナルが表示されます（スクロール表示でも同様に記録されますが、お使いのバッファ容量によっては、消え去ってしまう場合もあります）。
- 最近、変更されたメッセージおよびシグナルは灰色で強調されます。
- 少なくとも一回バス上に現れ、その後ある一定の長期間経過後に現れたメッセージおよびシグナルは、青で強調され、徐々にフェードアウトしてゆきます。

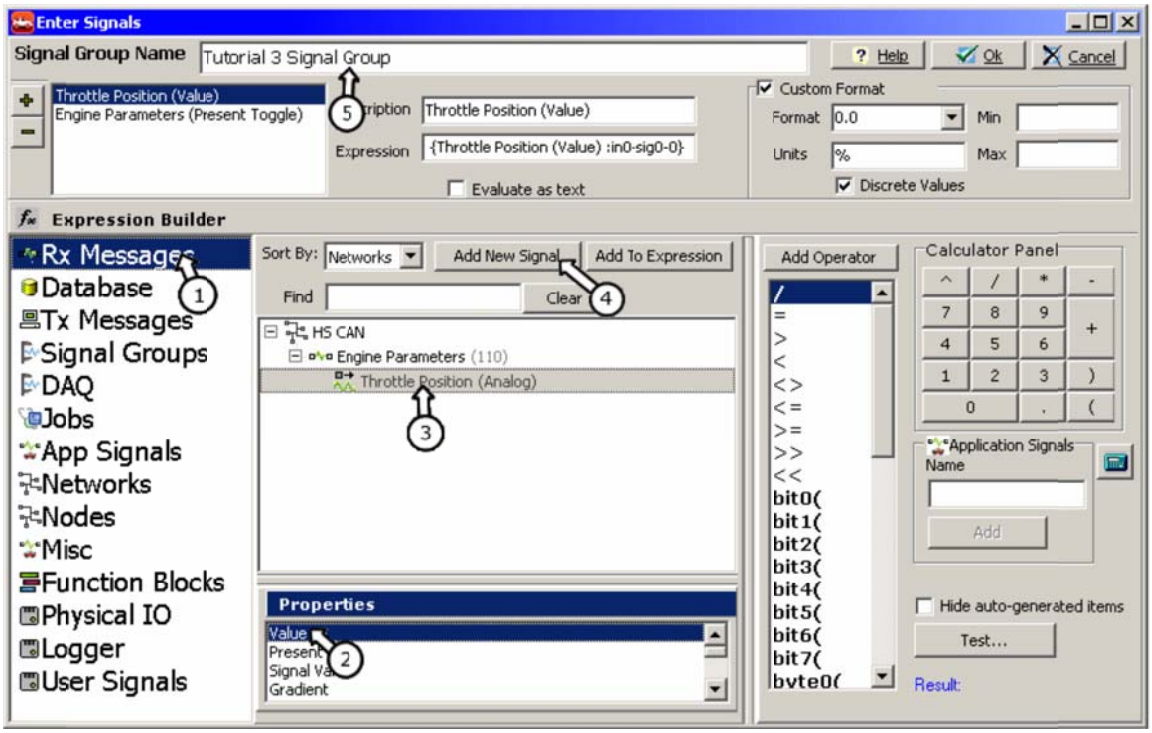
パート５：Signal List と Signal Plot からメッセージデータを観る

Vehicle Spy には、時々刻々と変化するシグナルを観測するための各種ツールを用意しています。Signal List は、表形式でシグナルデータの最新の値を示します。Signal Plot は、シグナルデータをグラフ表示します。下図は、複数のワークスペースを使用して開いた両方のビュー（上部に Signal plot、下部に Signal List）を示します。

1) **シグナルビューを開く：**Signal Plot と Signal List は、メニューバーの Measurement の下にあります。どちらのビューでも、Signal Group と呼ばれるシグナルのグループを作成することができます。



2) **Signal Group の作成：**Signal Group を作成するには、Signals Plot または Signals List ビューの Select Signals ボタンを押します（①）。Vehicle Spy Expression Builder が開きます。ここから、シグナルの選択をします。



3) **シグナル源の選択：**各 Signal Group は、複数のメッセージとシグナルを含めて、定義することができます。また、それらメッセージ及びシグナルの、複数のプロパティと値も含むことができます。シグナル源を選択して、開始して下さい（①）。Rx Messages を選択すると、Messages Editor で定義したすべての受信メッセージとシグナルが現れます。

4) **メッセージとシグナルの選択：**中央のネットワーク/メッセージ/シグナルのツリー表示されたエリアで、表示したいメッセージかシグナルを選択します（③）。次に、Properties エリアで、表示したいプロパティをハイライトします（②）。最後に、Signal Group に追加するために、Add New Signal ボタンをクリックします（④）。

この例では、Engine Parameters メッセージのプロパティである Present Toggle と、Throttle Position シグナルのプロパティである Value をグラフ化します。

5) **Signal Group に名前を付ける：**Signal Group の Name 欄をクリックし、Tutorial 3 Signal Group と入力します（⑤）。これは、複数の Signal Group を扱っている場合に重要です。

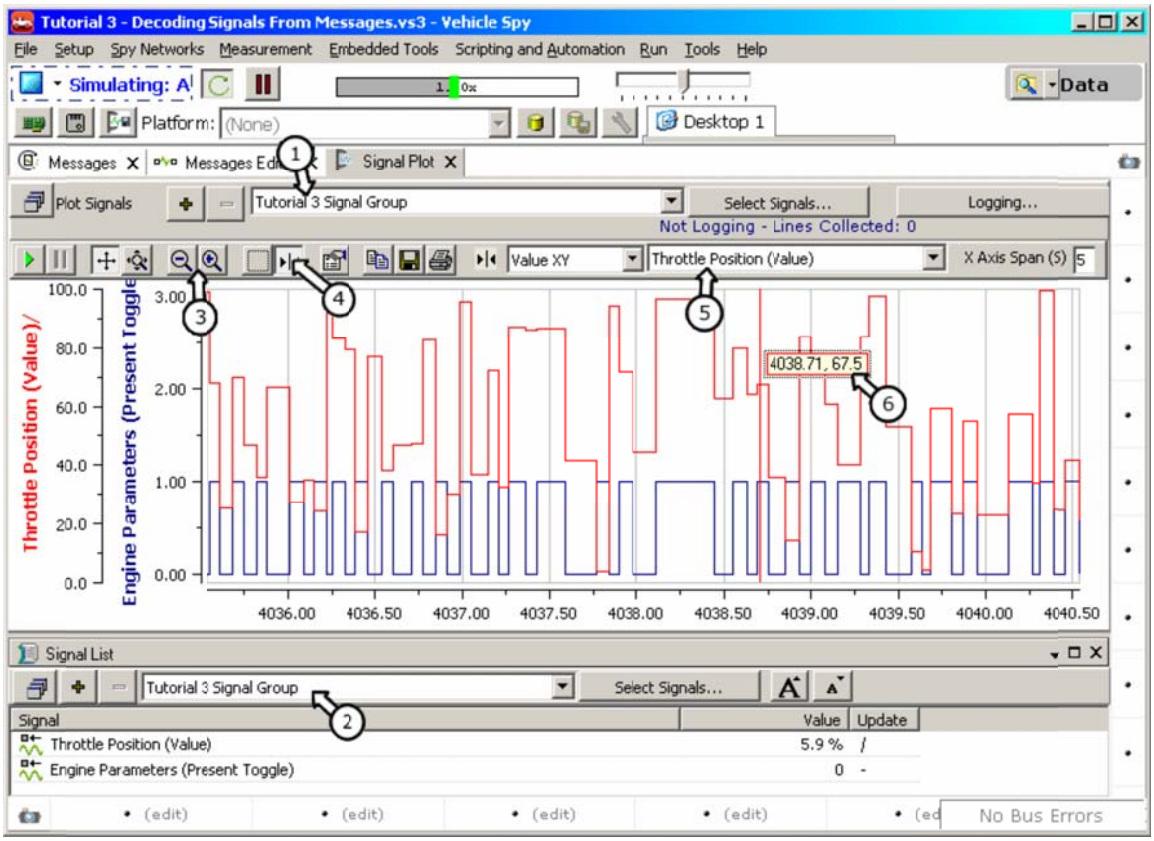
OK をクリックしてシグナルビューに戻る前に、Signal Group Name の直下のエリアの内容を再確認して下さい。このエリアには、Signal Group に含まれるすべてのデータが表示されており、先に指定した各プロパティは、カッコで表示されています。＋と－は Signal Group からアイテムを追加、削除するために使用します。例えば、あるアイテムを削除するには、それをクリックし、－ボタンを押して下さい。

パート６：Signal Plot 入門

Signal Plot は、非常に強力かつ多彩な機能を持つツールであり、その詳細は本チュートリアルの範疇を超えます。しかし以下では、そのいくつかの主要な機能について紹介します。

1) **表示する Signal Group の選択：**Signal Plot と Signal List は、同時に 2 個の異なる Signal Group を表示可能です。各ビューの上部にあるドロップダウンを使用して、Signal Group を選択して下さい（①&②）。

選択された Tutorial 3 Signal Group で、2 個の Y 軸が見えるはずです。それらは、違う色でグループのシグナルかメッセージに相当します。X 軸は、時間を表します。



2) **ツールバー上の各種ツールを試す：**Signal Plot グラフの上部にツールバーが見えます。ここには、このビューを操作するために必要なツール（Play, Pause, Mousewheel Axis Control, Zoom, Cursors, Setup, Save, Print）が含まれます（③&④）。

3) **カーソルとカーソル制御：**殆どのツールはその見た目で意味が掴めますが、カーソルツールは幾分の説明が必要です。ツールバーの Cursor アイコンをクリックして下さい（④）。するとグラフ上に、線と、それに隣接するボックスが現れます。この線をカーソルと呼び、測定に用います。ツールバーの右にみえる 2 つのドロップダウンは、カーソル制御に利用し、そのうちの右側のドロップダウンは、測定対象のチャンネルを選択する場所です（⑤）。

ドロップダウンから Throttle Position を選択して下さい。するとカーソルが、Throttle Position の Y 軸と同じ色に変わります。

一方、左側のドロップダウンは、測定対象の値を決める場所です。X 値/Y 値/X 値及び Y 値/ X 値の差/Y 値の差、のいずれかを選択可能です。

Value XY オプションを選択して下さい。すると、カーソルの隣のボックスに、X および Y の値が表示されます（⑥）。

以上が、簡単な Signal Plot の紹介です。本機能に興味を抱いた方は、ぜひ自身で色々と操作して試してみて下さい。

例題

以上のように、メッセージからのシグナルのデコードは、決して難しくはありません。本チュートリアルでは、以下を習得しました。

- メッセージの作成とシグナルのデコード
- 式の入力とテスト
- Messages ビューにおけるシグナルの表示
- シグナルビューでシグナルの表示

チュートリアル 4A：キャプチャ・ファンクションブロックとグラフィカルパネル

概要

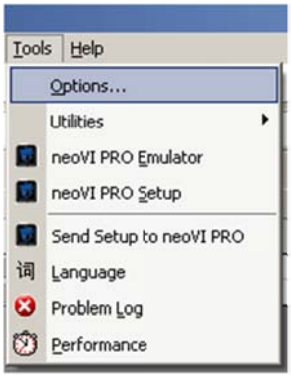
このチュートリアルでは、キャプチャ・ファンクションブロックの作成と使用、および、キャプチャ・ファンクションブロックを制御するためのグラフィカルパネルの作成方法を説明します。キャプチャ・ファンクションブロックは非常に柔軟で、さまざまな形で使用することができますが、今回は簡単のため 1 つの方法のみを紹介します。また、キャプチャ・ファンクションブロックによってキャプチャされたデータを手動で保存するための、ボタンを用いた単純なグラフィカルパネルの構成方法も習得します。

パート 1：シミュレーションの設定

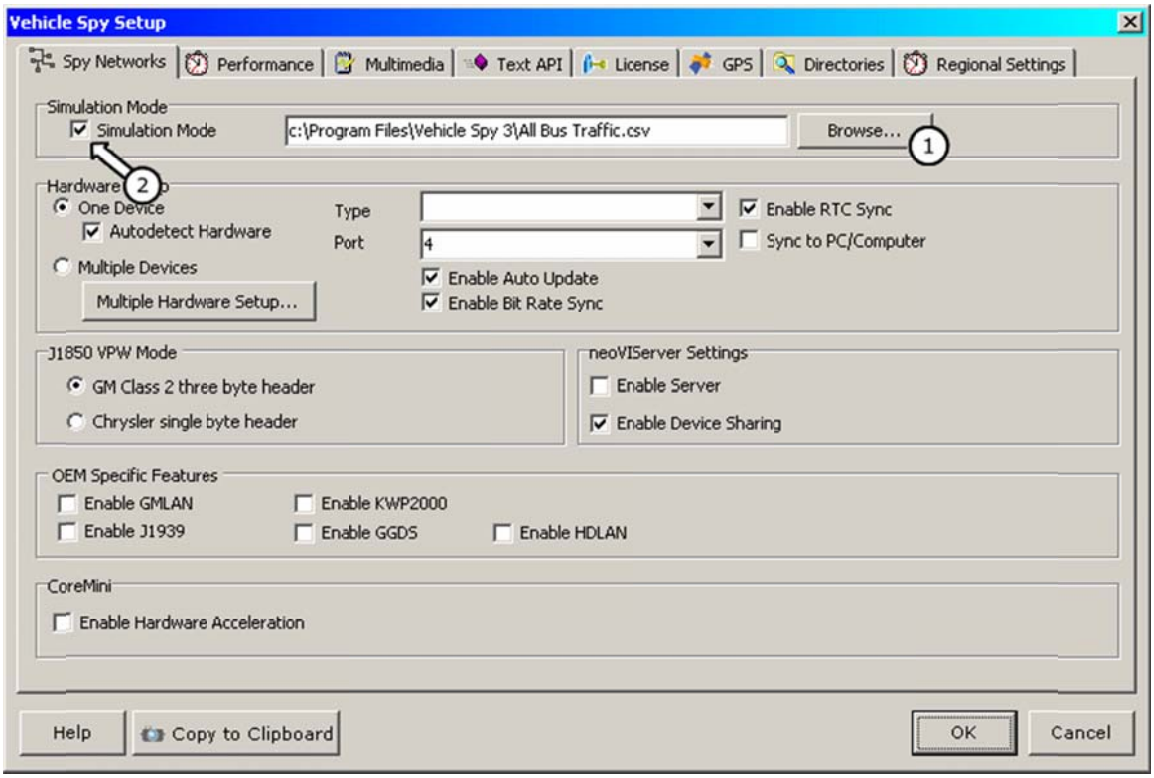
この例では、シミュレーションファイルを利用します。これにより、実際のバスに接続することなく、データキャプチャを実行できます。シミュレーションファイルとは、実際のバストラフィックをシミュレートするため、過去にキャプチャされたデータのことです。

1) **Vehicle Spy の開始**

2) **Options メニューを開く：**オプションメニューを開くには、Tools -> Options をクリックします。



3) **シミュレーションファイル（.CSV）を探す：**ブラウズボタンをクリックして、All Bus Traffic.csv を見つけて下さい（①）。これは Vehicle Spy 3 の Data Directory にあり、通常、C:\Program Files\Vehicle Spy 3\Data Directory または C:\Program Files (x86)\Vehicle Spy 3\Data Directory にあります。次に、Simulation Mode の横のボックスをクリックし、ウィンドウの下部の OK を押します。



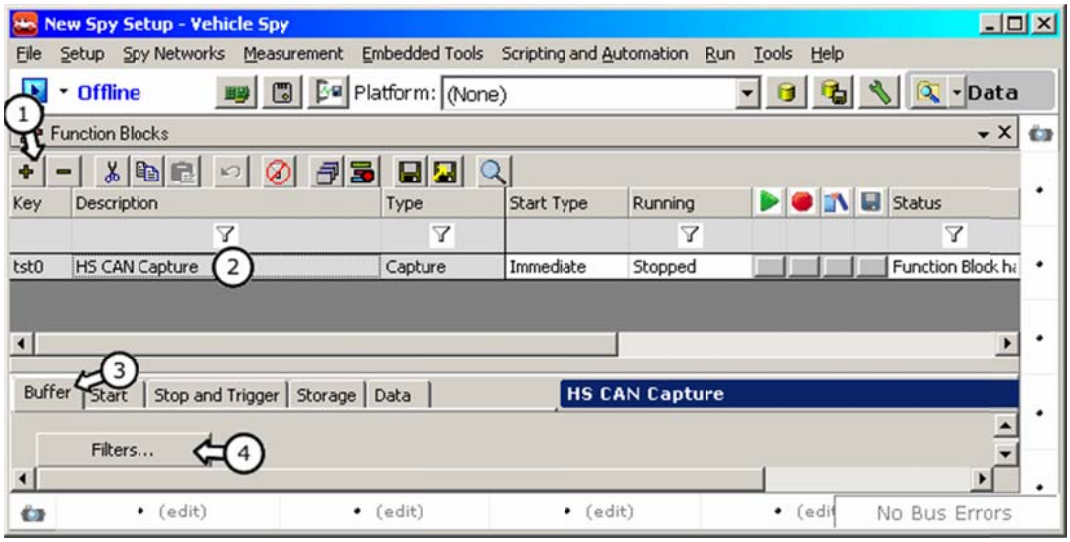
4) **シミュレーション開始：**これで、シミュレーションファイルが設定できました。画面左上の青いスタートボタンをクリックすると、シミュレーションファイルが再生されます。Messages ビューが開いていない場合、Spy Networks -> Messages をクリックして下さい。

パート２：キャプチャ・ファンクションブロックの設定

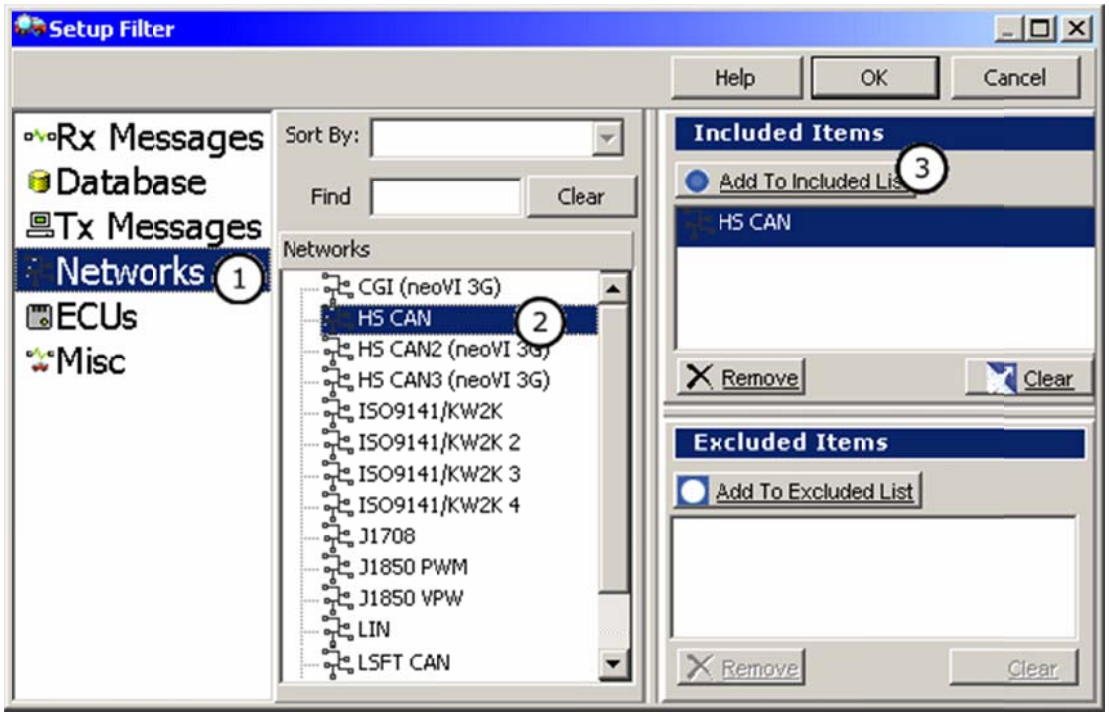
キャプチャ・ファンクションブロックにより、バスを流れるデータを、多彩な方法によりキャプチャすることができます。このチュートリアルでは、シミュレーションファイルからのデータをキャプチャします。

1) **Function Blocks タブを開く：**Scripting and Automation -> Function Blocks を選択し、Function Blocks タブを開きます。

2) **キャプチャ・ファンクションブロックの作成：**"＋"ボタンを押し、ドロップダウンから Capture を選択します（①）。Description セルをダブルクリックし、そのファンクションブロックを HS CAN Capture と命名します（②）。

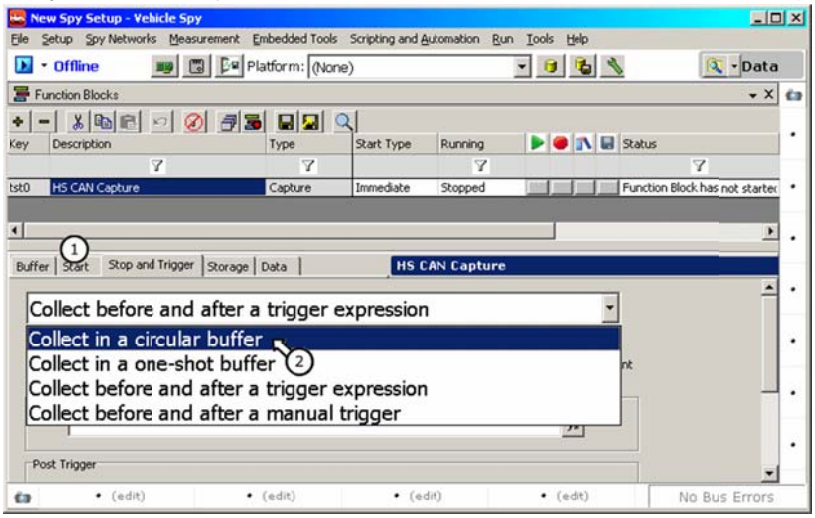


3) **フィルタの追加：**フィルタにより、キャプチャすべき対象を設定します。これは"Buffer"タブ(③)を開き、"Filters..."ボタンをクリックして設定します。すると、以下の画面が現れます。



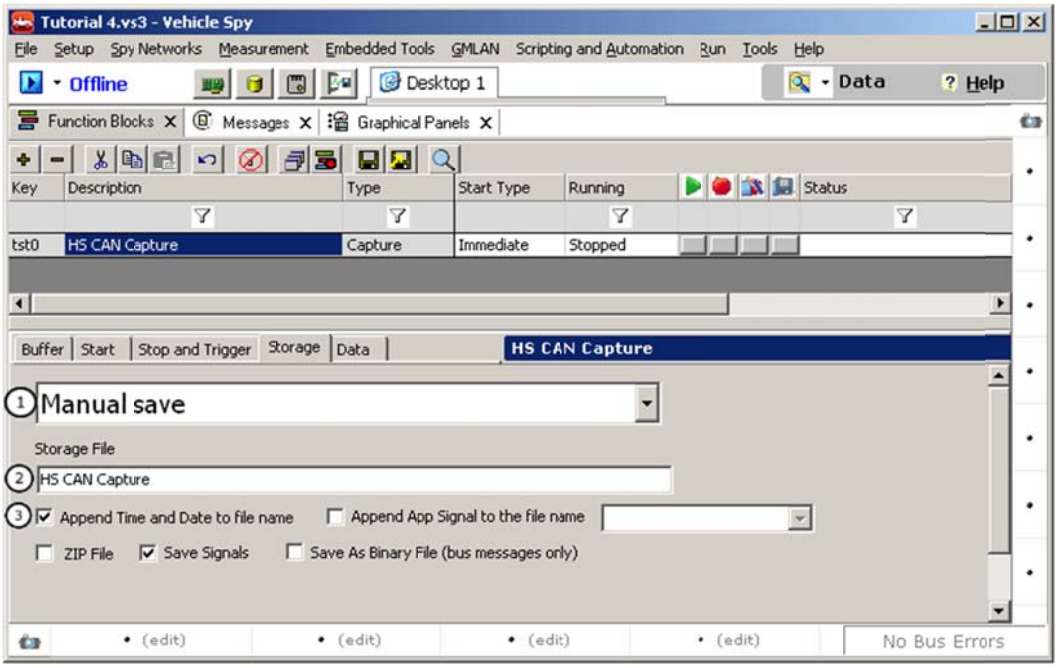
HS CAN のみのフィルタを作成するには、Networks(①)から HS CAN(②)を選び、”Add To Included List” ボタンをクリックして下さい。OK ボタンを押して、ウィンドウを閉じます。

4) **キャプチャ・ファンクションブロックの設定：**Start タブ（①）をクリックし、ファンクションブロックが”Start Immediately”に設定されていることを確認して下さい。次に、”Stop and Trigger”タブをクリックし、ドロップダウンから Collect in a circular buffer を選択します（②）。バッファサイズに 500 を入力して下さい。これはおよそ、シミュレーションファイル中の HS CAN メッセージ数に対応します。



5) **Storage オプションの設定**：Storage タブをクリックし、一番目のドロップダウンから、Manual slave を選択します（①）。ここで、様々な保存方法が選択可能です。Storage File の名前を、HS CAN Capture と変更してみましょう（②）。”Append Time and Date to file name”は、チェックを外し、無効

化して下さい（③）。

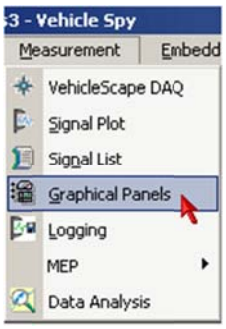


6) **2 つ目のキャプチャ・ファンクションブロックの設定：** 2 つ目のキャプチャ・ファンクションブロックを、MS CAN Capture と名付けて作成して下さい。これは、バッファサイズが 370 であることを除いて、先のファンクションブロックと同様に設定して下さい。

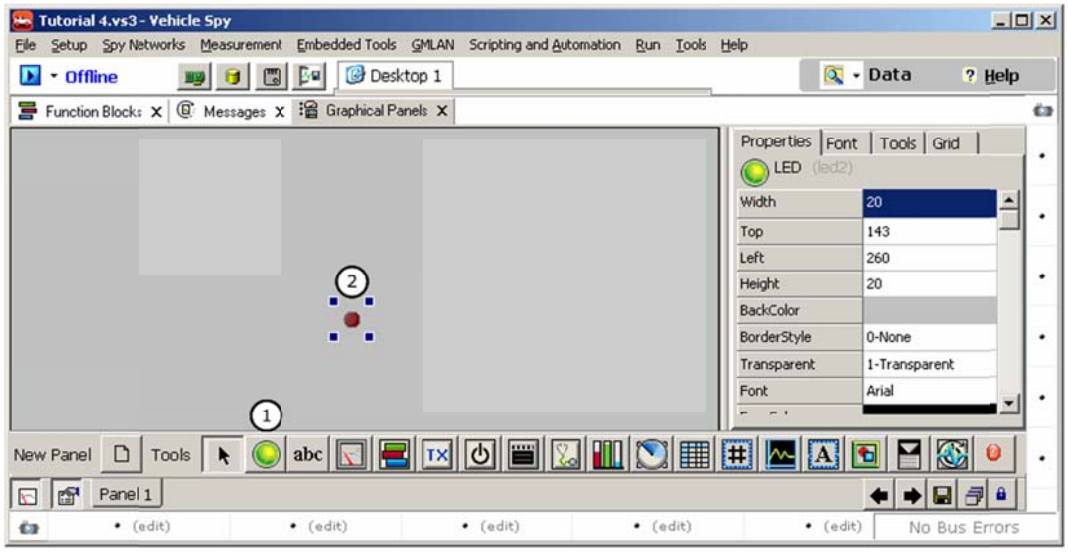
パート３：グラフィカルパネル

グラフィカルパネルは、アプリケーションの GUI に似ています。一度作成すると、Vehicle Spy に関する知識を持たないユーザでも、Vehicle Spy 上での複雑な作業が可能になります。このチュートリアルでは、少数のツールを含む、極めて基本的なグラフィカルパネルを構成します。複数パネルに及ぶ、何十ものツールを含むシステムの構築も可能です。

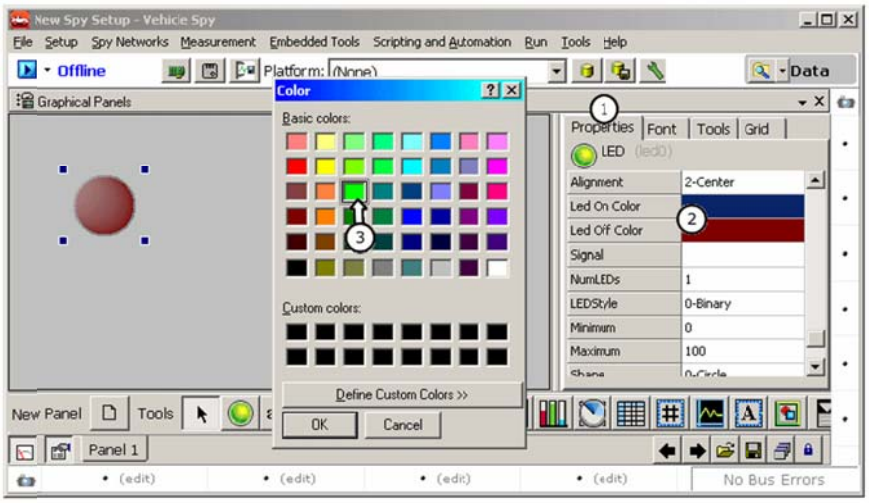
1) **Graphical Panels タブを開く：**Measurement -> Graphical Panels をクリックし、Graphical Panels タブを開きます。



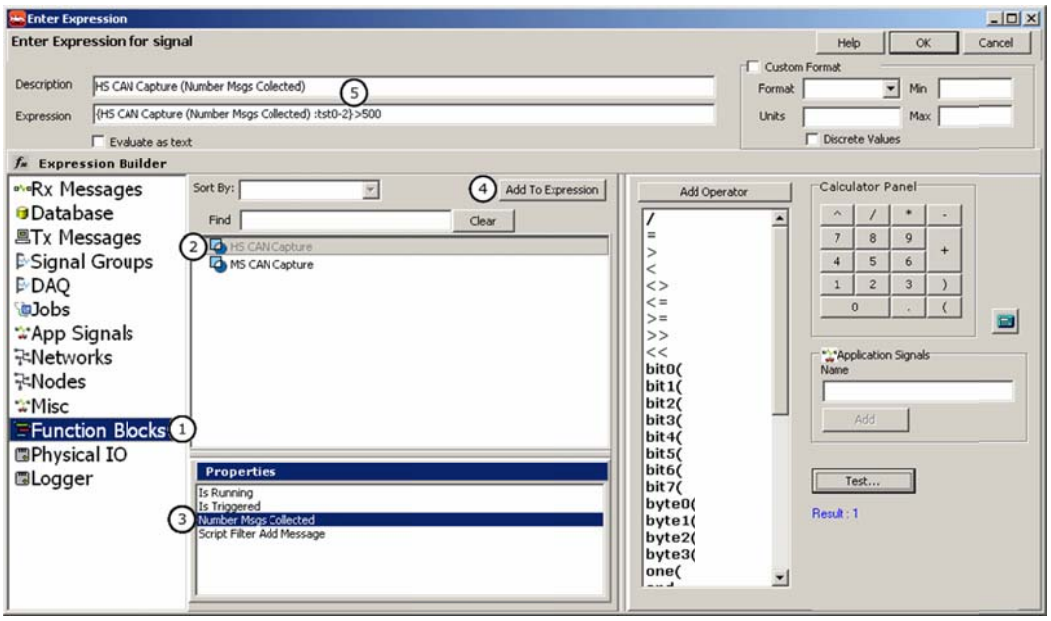
2) **LED の作成：**ツールを作成するには、パレット上から単にツールをクリックし、グラフィカルパネル中の所望の位置を再度クリックして、ツールを配置します。まず、画面下部の LED ツール（①）をクリックし、 続けてグラフィカルパネル中の適当な場所をクリックしてみましょう。LED がグラフィカルパネル内に現れるはずです（②）。LED の周囲の青い囲いをドラッグすることで、LED のサイズ調整ができます。



3) **LED の設定：**先の LED をクリックします。右側の Properties ウィンドウは、LED の設定用オプションを含みます。もし Properties タブが見えなければ、LED と Properties タブが選択されているかを確認して下さい（①）。Properties ウィンドウから、Led On Color プロパティ（②）をクリックし、続けてダブルクリックして色選択画面を呼び出して下さい。ここでは、LED の色を緑とします(③)。

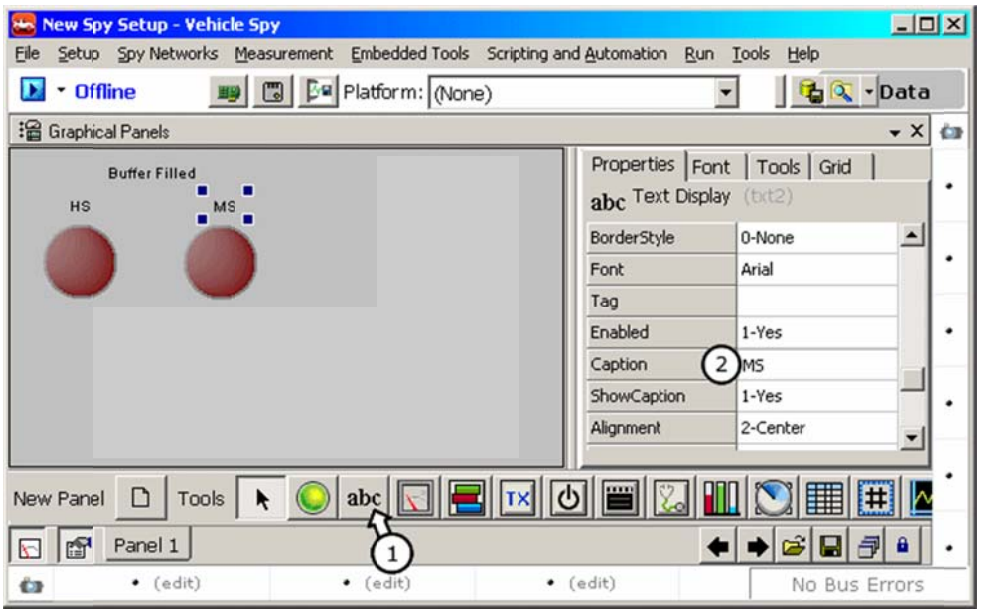


Signal プロパティは、そのツール(この場合は LED)と、Vehicle Spy 上のデータを関連付けるものです。Signal プロパティをダブルクリックして、Expression Builder を開いて下さい。Expression Builder ウィンドウから、Function Blocks をウィンドウ左側から選びます（①）。すると、画面中央(②)に、これまで作成したファンクションブロックがリストされます。HS CAN Capture を選択し、その下の Properties 画面から Number Msgs Collected を選択して下さい（③）。選択したアイテムを Expression 欄に入力する方法にはいくつかあります。１つは、そのアイテムをダブルクリック、もしくは Add To Expression ボタンを押下することです（④）。Expression 欄に追加後、”>500”(シミュレーションファイル中の、およその HS CAN メッセージ数)を⑤のように追加して下さい。これにより、収集したメッセージ数が 500 以上となると、LED が点灯します。

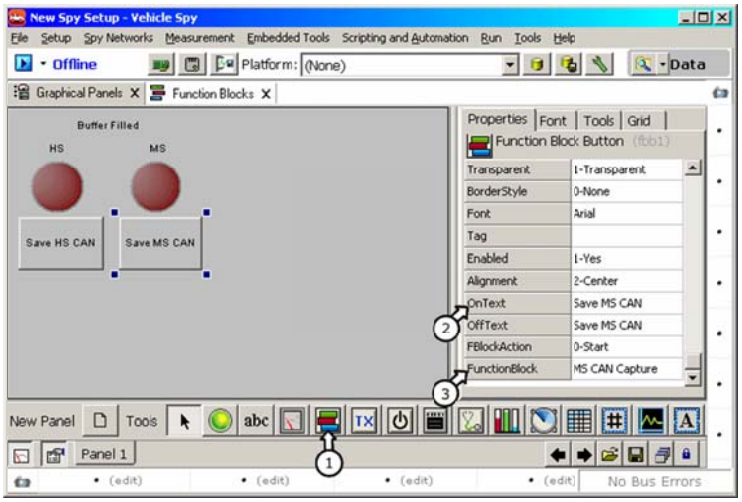


4) **2 つ目の LED を構成：**上記の方法で、2 個目の LED を構成して下さい。ただし、MS CAN Capture の Number Msgs Collected プロパティを使用し、”>500” の代わりに” >300” を使用します。

5) **Text Display ツールの作成：**このツールは、テキスト値の表示に利用されますが、単なるラベルとしても使用可能です。このチュートリアルでは、LED のラベリングのために使用します。Text Display（①）をクリックし、グラフィカルパネル上の適当な場所をクリックします。Text Display をラベルとして使用するには、Caption を所望の内容に変更します（②）。ここでは、Buffer Filled とし、LED の上にテキストを配置して下さい。また各 LED に対して 1 つずつ、HS CAN と MS CAN 用のラベルを作成します。

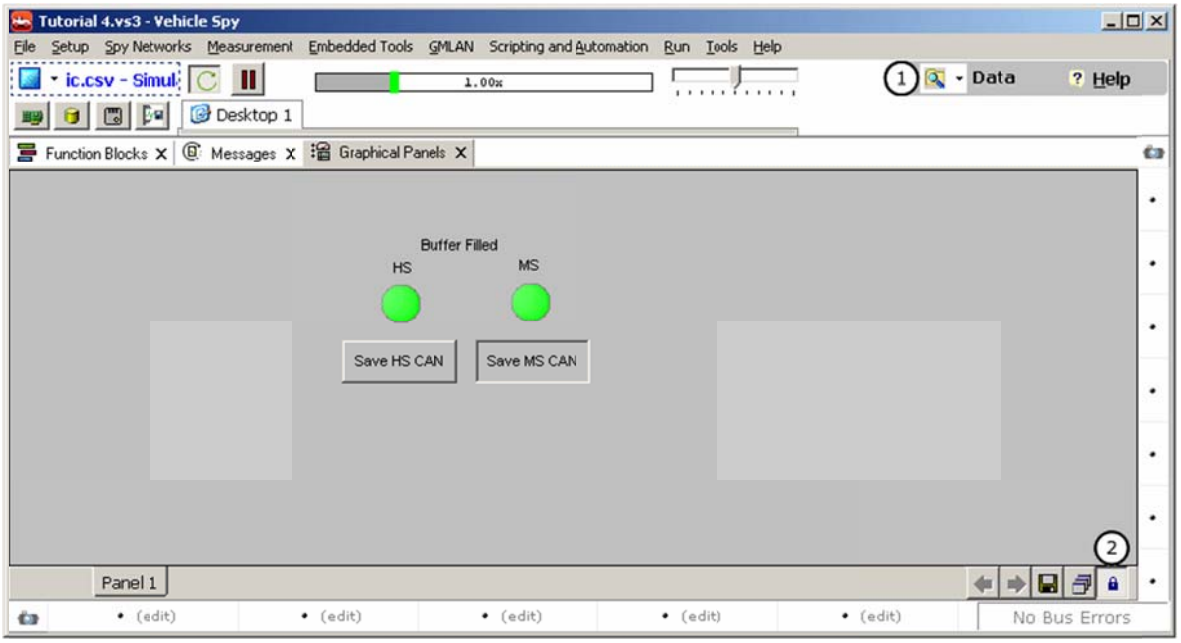


6) **Function Block Button の作成：**他のツールと同様に、Function Block Button（①）をクリックし、グラフィカルパネル内に置きます。OnText と OffText プロパティを"Save HS CAN"(②)へと変更して下さい。次に、FBlockAction プロパティをダブルクリックし、3-Save を選びます。これにより、Function Block Button を押下すると、対応するファンクションブロックが、収集したデータを保存するようになります。FunctionBlock プロパティをダブルクリックし、HS CAN Capture を指定して下さい。最後に、２つ目の Function Block Button を作成して下さい。今回は、OnText と OffText を"Save MS CAN"へと、また Function Block としては MS CAN Capture を指定して下さい。



パート４：まとめ

作成したファイルをテストするため、画面左上、青色のスタートボタンをクリックして下さい。グラフィカルパネルを観察すると、バッファ中のデータが指定の量に達するまでは赤色、それ以後は緑色へと点灯するはずです。作成した２つのセーブボタンをそれぞれ押して、バッファをデータディレクトリへと保存してみましょう。データディレクトリボタン（①）をクリックすると、保存場所を参照できます。またグラフィカルパネルは、画面右下のロックボタン(②)を押下することで、ボタン類の位置を固定して動かないようにすることが可能です。



以上で完了です！以下の例題で、これまで学んだ内容を補強しましょう。例題２は新たな概念を含みますが、5.7.1.4.2 キャプチャ型の項目を参考にして下さい。

例題

１．ISO9141/KW2K 用のキャプチャ・ファンクションブロック、LED、Function Block Button を追加して下さい。ヒント：１ループあたり、145 のメッセージが流れます。

２．one-shot buffer にデータを格納するキャプチャ・ファンクションブロックを作成し、さらに、それが Function Block Button によって開始されるように設定して下さい。ヒント：FBlockAction プロパティの使用。また、Start Immediately の代わりに Manual Start を使用して下さい。

チュートリアル 4B：スクリプト及びプレイバック・ファンクションブロック

概要

このチュートリアルでは、ファンクションブロックの作成及び使用方法を説明します。ファンクションブロックは、Vehicle Spy と neoVI ハードウェアの処理を自動化するための強力なツールです。このチュートリアルでは、基本的なグラフィカルパネルの作成に加え、スクリプト・ファンクションブロックとプレイバック・フ

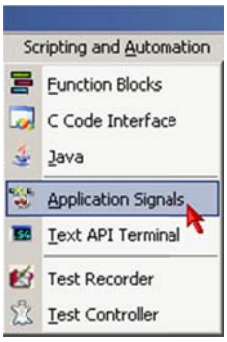
ァンクションブロックの作成を行います。なおこのチュートリアルには、Tutorial Script Files.zip が必要です。

パート１：アプリケーションシグナル

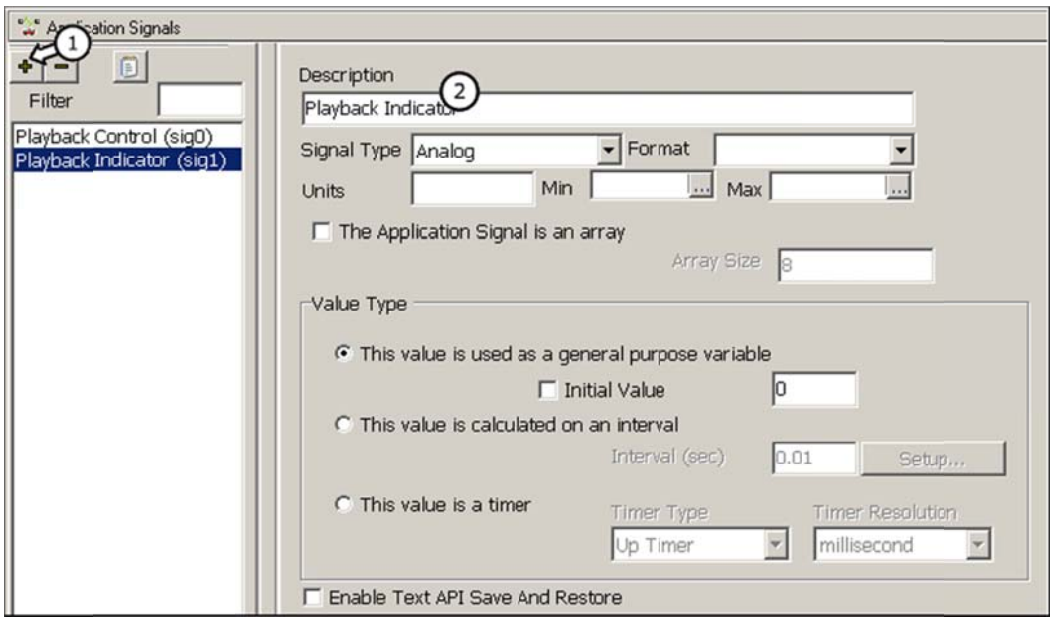
アプリケーションシグナルは、プログラミング言語で用いられる、プログラム内にデータを格納する「変数」に似ています。Vehicle Spy においても同様、アプリケーションシグナルはデータの格納に利用されます。本チュートリアルで、LED の切り替え、およびボタンの状態の保存のために、アプリケーションシグナルを使用します。

1) **Vehicle Spy の開始**

2) **Application Signals タブを開く：**Scripting and Automation -> Application Signals を開きます。

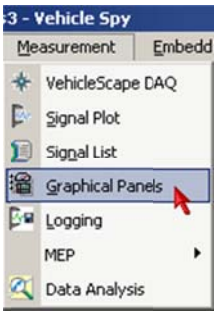


3) **アプリケーションシグナルを作成：**左上の＋を押してアプリケーションシグナルを作成します（①）。このアプリケーションシグナルに、Playback Control という記述文をつけてみましょう（②）。そして、Playback Indicator という名の別のアプリケーションシグナルを作成します。

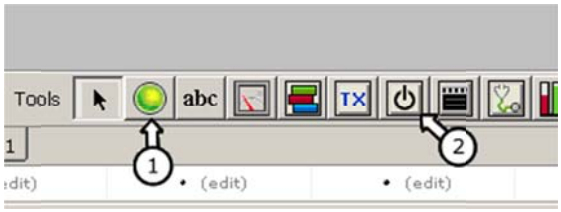


パート２：グラフィカルパネル

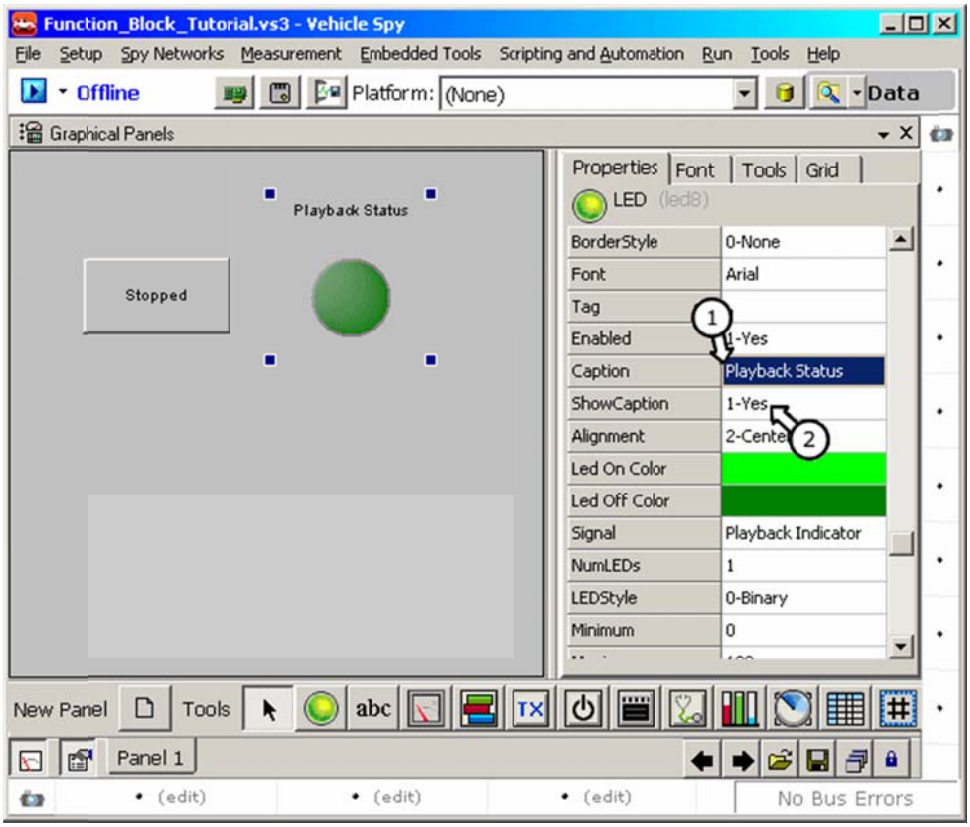
1) **Graphical Panels タブを開く：**Measurement -> Graphical Panels をクリックして下さい。



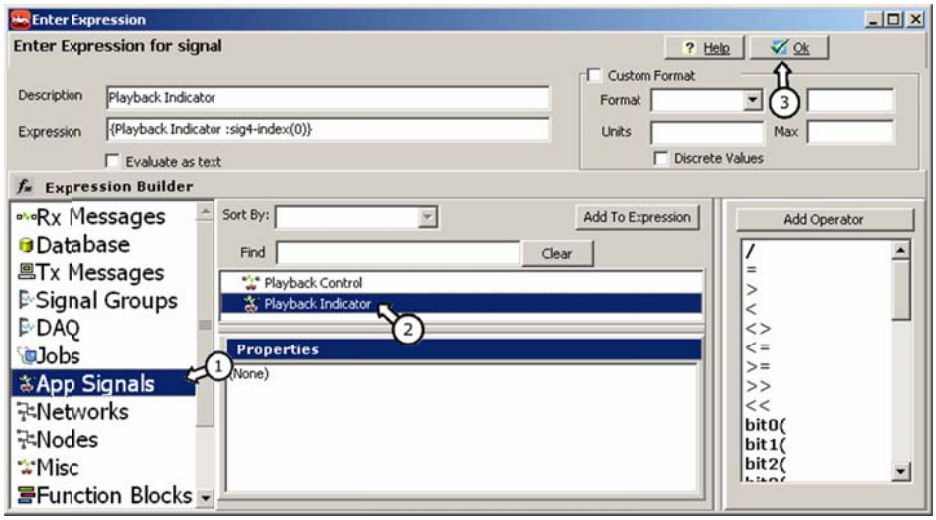
2) **必要なツールを作成：**画面の下部の緑の丸をクリックして、LED を作成します（①）。そしてグラフィカルパネル内の適当な場所をクリックし、LED を配置します。次に、②で示す On/Off ボタンを押し、グラフィカルパネル内に配置して下さい。



3) **ツールのプロパティを変更：**グラフィカルパネルに置いた LED をクリックし、画面右側にプロパティを表示して下さい。この中の Caption プロパティを、Playback Status に変更します。次に、Show Caption プロパティをクリック、Yes（②）を選択して下さい。必要に応じて LED のサイズを変更します。
On/Off ボタンをクリックし、OnText プロパティを Running に、OffText を Stopped に変更します。
On/Off ボタンの表示文は、それが On か Off に設定されているかによって変わります。ボタンを実際にクリックしてみればわかるでしょう。

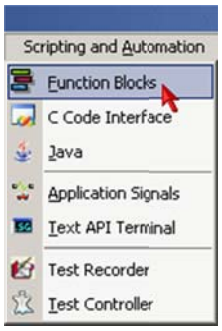


4) **アプリケーションシグナルにツールを対応付ける：**LED をクリック、画面右側の Signals プロパティをダブルクリックして、Expression Builder が開きます。左側にみえる、アプリケーションシグナルのカテゴリを選択して下さい（①）。Playback Indicator シグナル（②）をダブルクリックします。シグナルをダブルクリックすると、それ自身が Expression フィールドに入ります。ここで、変更を確定するために OK を押します（③）。デフォルトで、LED が割り付けられるシグナル値が 1 と等しいとき、LED が点灯します（LED を制御する他の方法もありますが、ここでは割愛します）。On/Off ボタンに関しても同様に設定して下さい。On/Off ボタンを押すことで、アプリケーションシグナルの値が、1(On)または 0(Off)に切り替わります。

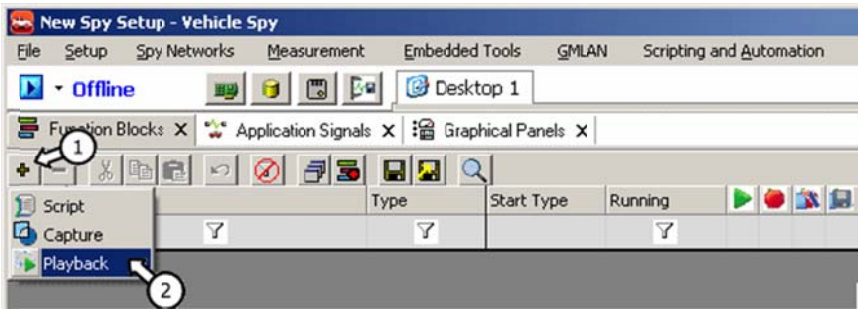


パート３：プレイバック・ファンクションブロック

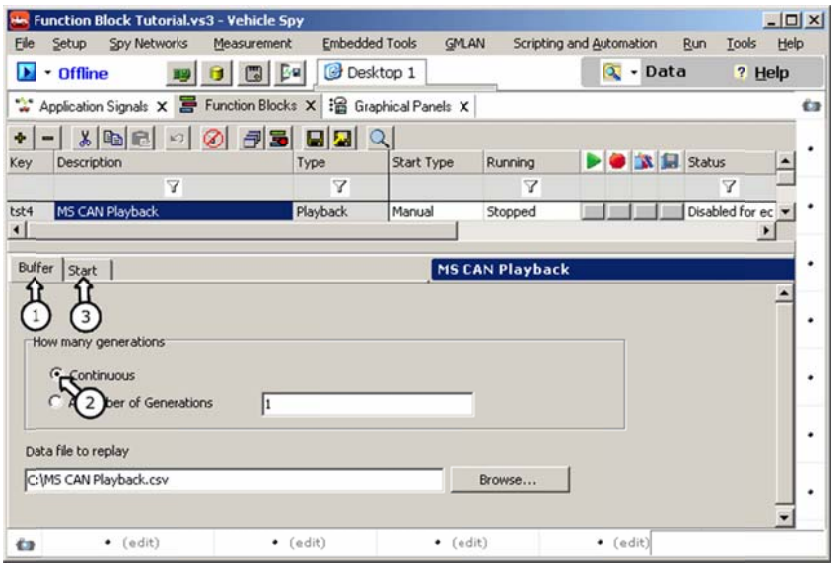
1) **Function Blocks タブを開く：**Scripting and Automation -> Function Blocks をクリックして下さい。



2) **プレイバック・ファンクションブロックを作成：**①に示す＋ボタンをクリックし、ドロップダウンから Playback を選択します（②）。Description 欄をダブルクリックし、ファンクションブロックに名前をつけて下さい。ここでは、MS CAN Playback としましょう。



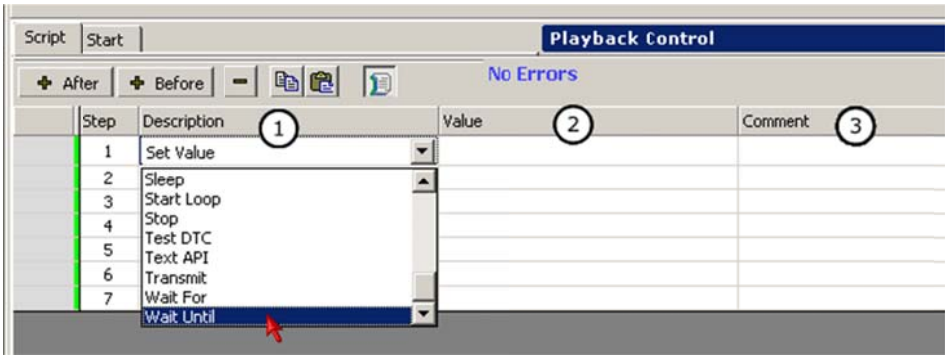
3) **プレイバック・ファンクションブロックの設定：**Buffer タブ（①）を選択し、Continuous プレイバックが有効化されていることを確認します（②）。”Browse...”ボタンをクリックし、本チュートリアル冒頭で示した MS CAN Playback.csv ファイルを指定して下さい。最後に、Start タブ（③）をクリックし、最初のドロップダウンから Manual Start を選択します。



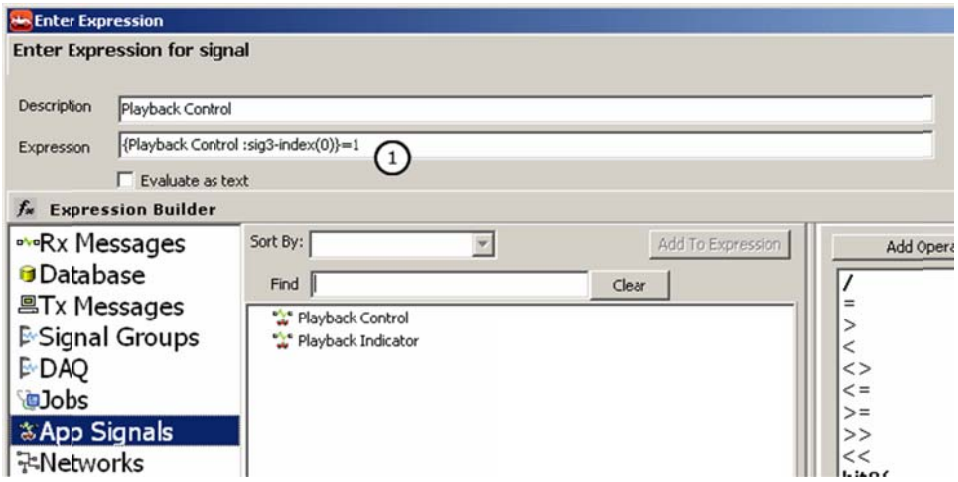
パート４：スクリプト・ファンクションブロック

1) **スクリプト・ファンクションブロックの作成：**これまでのプレイバック・ファンクションブロックと同様の方法で作成します。今回はドロップダウンから、Script を選択し、Playback Control という名前をつけて下さい。

2) **スクリプト・ファンクションブロックの設定：**Description セル（①）をダブルクリックすることで、新しいステップを追加できます。Description とは、実行したいコマンド内容を示します。Value 欄(②)では、そのコマンドのパラメータを決定します。一例としては、Wait for の Description に対して、2.00 秒の Value 等があげられます。Comment 欄(③)は、各ステップの意味を忘れないための、備考欄です。コメントはスクリプトの動作には影響しないので、自由に使用して下さい。



--- ステップ 1　<u>Wait Until</u>:　Description 欄をダブルクリックし、ドロップダウンから Wait Until を選択して下さい。次に、Value セルをダブルクリックします。すると、Part 3 で Graphical Panel タブから開いたものに似たダイアログが開きます。App Signals の中の Playback Control をダブルクリックし、ウィンドウの上部の Expression フィールドをクリックします。カッコ{ } のあとに、=1 を入力して下さい(①)。このステップは、グラフィカルパネルの On/Off ボタンが押されるまで、処理を待ちます。



--- ステップ 2　<u>Function Block Action</u>: Function Block Action を選択し、Value としては Start を選択、さらに"MS CAN Playback"を選んで下さい。このステップにより、MS CAN バスのシミュレーションファイルが再生されます。

--- ステップ 3　Set Value:　Set Value を選択、Value としては App Signals を選択、さらに Playback Indicator をダブルクリックして下さい。Expression 欄には、単なる 1 を入力します。イコール記号は必要ありません。このステップにより LED が点灯し、シミュレーションファイルの再生が始まったことがわかります。

--- ステップ 4　Wait Until:　もうひとつ、Wait Until を作成します。今回は、Playback Control がゼロになるのを待つよう設定して下さい。

--- ステップ 5　Function Block Action:　再び Function Block Action を作成します。今回は、”MS CAN Playback” ファンクションブロックを”Stop”します。

--- ステップ 6　Set Value:　Playback Indicator をゼロに戻して下さい。

--- ステップ 7　Jump to:　最後に、ステップ 1 に戻るための Jump to を作成します。Value セルをダブルクリックし、1 を入力します。これにより、すべてのステップを完了した後、ファンクションブロックが再スタートします。

3)　最終結果：すべてのステップをスクリプトに追加すると、下図**スクリプト・ファンクションブロックの作成**
以前のページの Part 4 と同様な方法で、新しいファンクションブロック作成します。今回、ドロップダウンから Script を選択します。このファンクションブロックを Playback Control をいう名前をつけます。

4)　スクリプト・ファンクションブロックの設定
スクリプト・ファンクションブロックで新しいステップを作成するには、Description セル（①）をダブルクリックします。そして、ドロップダウンから希望のステップタイプを選択します。記述は、動作させるための命令です。次は、どのような命令をするかです。これは、Value セル（②）で行われます。値は、ステップの各タイプに特有です。ステップタイプと値の例としては、Wait for 記述と 2.00 seconds 値です。Comment セルは、各ステップでスクリプトが何を行うかを覚えておくためのコメント欄です。スクリプトは動作に影響しませんので、自由に使用して下さい。下図を参照して下さい。

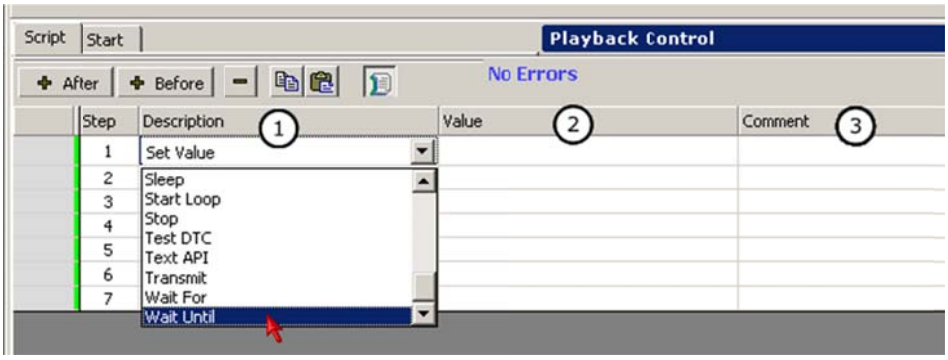


図 2.7.4-1　新しいスクリプトを開始

--- Step 1　Wait Until:　Description セルをダブルクリックし、ドロップダウンから Wait Until を選択して、Wait Until ステップを作成します。次に、Value セル（②）をダブルクリックします。

Value セルをダブルクリックした後、Part 3 で Graphical Panel タブから開いたものに似たダイアログが開きます。今回、App Signals の下の Playback Control をダブルクリックし、そして、ウィンドウの上部の Expression フィールドをクリックします。ブレース { } のテキストのあとに、=1 を入力します。このステップは、グラフィカル・パネルの On/Off ボタンが押されるまで待ちます。

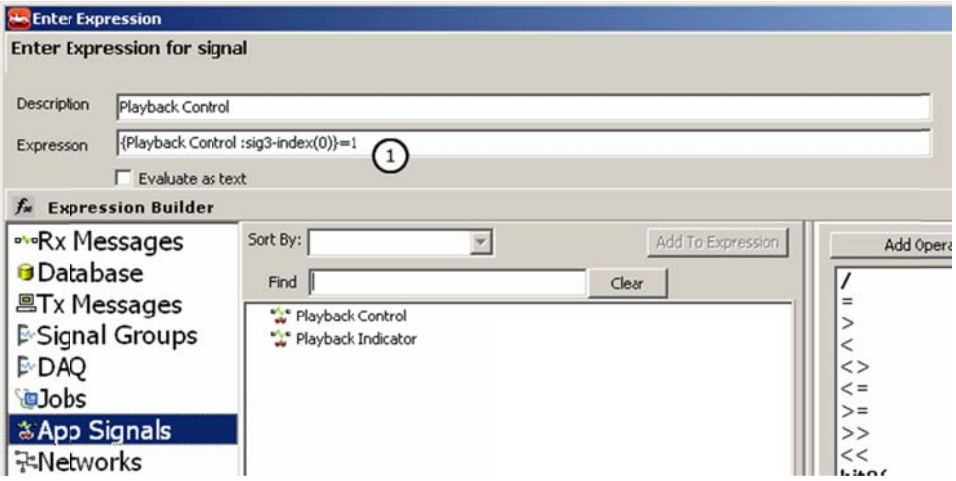


図 2.7.4-2　Expression Editor

--- Step 2　<u>Function Block Action</u>: ステップ 2 で、左からの Value 選択 Start に対する Function Block Action を作成し、そして、OK を押します。このステップでは、MS CAN をシミュレートしている Playback ファイルを開始します。

--- Step 3　<u>Set Value</u>:　次に、Set Value ステップを作成します。Value に対して App Signals を選択し、そして、Playback Indicator をダブルクリックします。ここで、ウィンドウの上部の Expression フィールドに 1 を入力します。Set Value は、別に動作し、そして Wait Until、その中で、等しいことが含まれ、必要なくなります。これは、Playback が始まったことを表し、LED が点灯します。

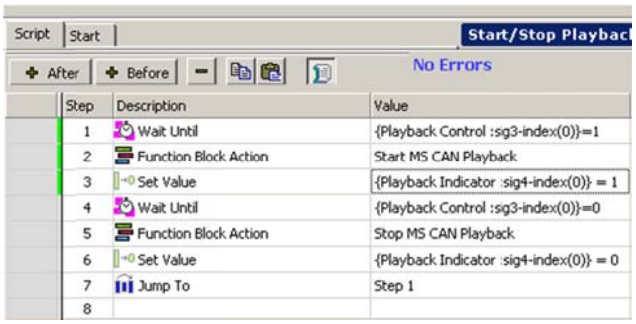
--- Step 4　<u>Wait Until</u>:　もうひとつ、Wait Until を作成します。今回、Wait Until は、Playback Control がゼロに等しくなるのを待ちます。

--- Step 5　<u>Function Block Action</u>:　次に、もうひとつの Function Block Action を作成します。今回、アクションは、"MS CAN Playback" ファンクションブロックを"Stop"します。

--- Step 6　<u>Set Value</u>:　この Set Value では、Playback Indicator をゼロに戻します。

--- Step 7　<u>Jump to</u>:　最後に、ステップ 1 にとぶ Jump to を作成します。Value セルをダブルクリックし、1 を入力します。これは、すべてのステップを完了した後、ファンクションブロックがリスタートすることを確実にします。

5)　**最終結果：**すべてのステップをスクリプトに追加すると、下図の様になるはずです。



注意:“sig”の後の数字は異なるかもしれません。sig の数は、インデックス用のものです。

パート５：まとめ

お疲れ様でした。ここで、Vehicle Spy をオンラインにしてみましょう（画面左上スタートボタン押下）。グラフィカルパネルから MS CAN Playback をオンにすると、MS CAN のトラフィックが見えるはずです。今回作成したプロジェクトファイルにより、実際のバス上にて、キャプチャした ECU のトラフィックを流し、その動作をシミュレートすることができます。

チュートリアル５：DTC データのリクエスト及び保存

概要

本チュートリアルでは、ファンクションブロックとグラフィカルパネルを用いて、Diagnostics Job サービスを利用する方法を説明します。最終結果は、Diagnostic Trouble Codes(故障診断コード)の要求及び保存を行う、グラフィカルパネル上のボタンになります。本チュートリアル完了後に出来上がるプロジェクトファイルの一例としては、Logon 画面の Example タブにある”DTCRequestAndSave”を参照して下さい。

本チュートリアルでは、以下のステップを通して説明します。

1. DTC リクエストサービスの作成
2. 情報を要求＆保存するファンクションブロックの作成
3. エンドユーザーのためのグラフィカルパネルの作成
4. ログデータの検索と参照

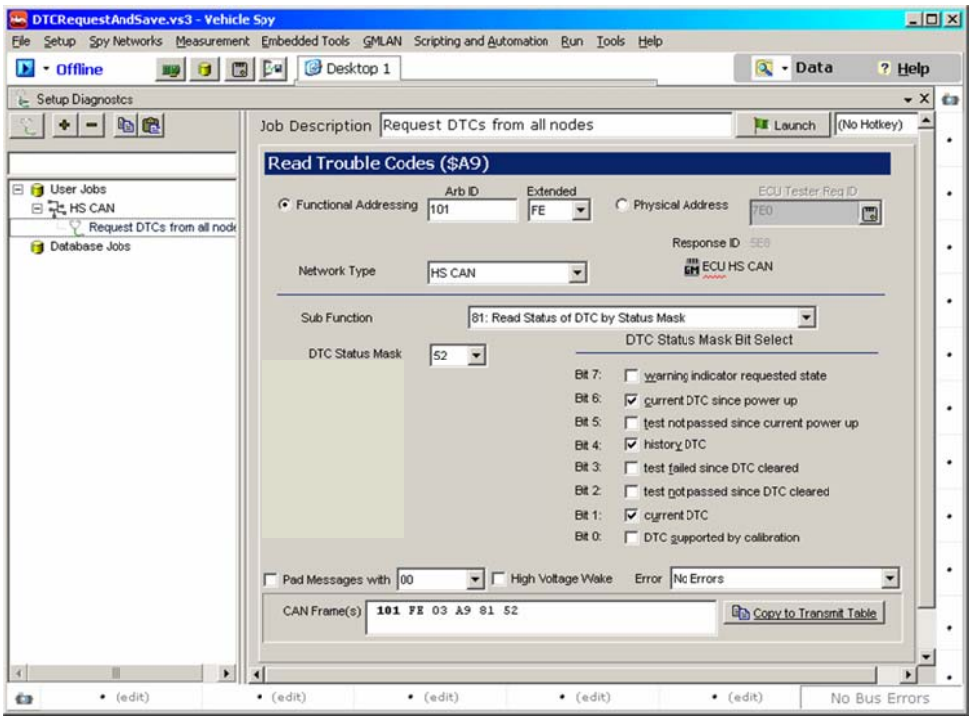
パート１：診断ジョブを作成

まず、DTC を要求するための、診断ジョブの作成を始めましょう。

1) **Vehicle SpyLogon**：Vehicle Spy を開き、Logon Name を選択した後、メイン・メニュー・バーから、File -> New を選択します。本チュートリアルでは、Logon Name は特に重要です。収集されたデータは、選択したユーザのデータディレクトリに保存されます。

2) **DTC 要求ジョブの作成**：次に、DTC を読み込むための診断ジョブを作成します。Vehicle Networks -> Diagnostics Setup から、Diagnostics 画面を開いて下さい。続けて、＋ボタンを押して GMLAN $A9 Check Codes ジョブを追加します（注：この例は、どの診断サービスでも動作します）。

3) **設定の確認と Description の編集**：この例では、デフォルト設定で問題ありません。この診断ジョブは、GMLAN 車両の High Speed CAN ネットワーク上で機能するすべての ECU から、DTC を要求します。Job Description を Request DTCs from all nodes に変更して下さい。設定は、下図のようになるはずです。



パート２：データを要求及び保存するファンクションブロックを作成

1) **Function Blocks ビューを開く**：次のステップは、Diagnostic サービスを制御し、データを保存するためにファンクションブロックを作成することです。Scripting and Automation -> Function Blocks を選択し、Function Blocks ウィンドウを開きます。

2) **ファンクションブロックの作成**：＋ボタンをクリックし、Script を選択します。ファンクションブロックの名前を、例えば Check Codes and Log Them to a File のような、分かりやすいものに変更して下さい。

3) **スクリプトの作成**：このタスクでは、4 ステップが必要です。必要なステップとその意味に関しては、

下記の表を参照して下さい。Diag Job Action の Value 欄をダブルクリックすると、診断ジョブの動作を選択できます。

ステップ	コマンド	Value	内容
1	Diag Job Action	Start Request DTCs from all nodes	診断サービスを開始します。
2	Diag Job Action	Wait until Complete Request DTCs from all nodes	選択した診断サービスが終了するまで、ここで待機します。サービス終了後、次のステップに進みます。
3	Diag Job Action	Save Request DTCs from all nodes	受信したデータをログファイルへと保存します。
4	Stop	n/a	ファンクションブロックを終了します。

4) **スタートの設定：**Start タブで Manual Start を選択します。これにより、スクリプト・ファンクションブロックの、次のパートで設定するグラフィカルパネルのツールによる開始が可能となります。

パート３：グラフィカルパネルを作成

グラフィカルパネルを構成することで、エンドユーザにとってより簡単なトラブルコードの要求を行います。

1) **グラフィカルパネルを開く：**Measurement -> Graphical Panels を選択して下さい。基本的に、グラフィカルパネルの作り方は各人の自由です。創造性をいかんなく発揮して下さい！

2) **パネルを構成：**本パネルでは、３つのツールを利用します。

最初は、ファンクションブロックを開始するための Function Block Button です。設定すべきボタンのプロパティは、
FunctionBlock：“Check Codes and Log Them to a File”
(先に自身で作成したファンクションブロック名)
FBlockAction：0-Start

これにより、ボタンがクリックされるごとにそのファンクションブロックが開始されます。

次に、トラブルコード要求がまだ進行中であることをユーザに通知するために、 LED を追加してみましょう。LED を作成し、下記のプロパティを設定します:
Signal：Jobs カテゴリの中の、Is Running プロパティを持つ診断ジョブを選択します。

最後に、Text Display の Caption プロパティを使用して、各ボタンに簡単な説明文をつけてみましょう。下図に完成例を示します。

パート４：DTC の要求、保存、表示

1) **パネル作動：**ハードウェア[11]を、HSCAN ネットワークを持つ車両または ECU に接続し、Vehicle Spy を開始して下さい。グラフィカルパネルのボタンをクリックし、作成したファンクションブロックを開始します。LED の色が変わるはずです。その色が元に戻ったとき、収集したデータがファイルに保存されます。

2) **保存したデータを参照：**データの保存先は、右上の Data ボタンをクリックすると現れます。これは、パート 1 で設定した Logon Name に対応するデータフォルダになります。データフォルダに保存されたファイルの名前は、診断サービス名に続き、日付・時刻を持ちます（例：Request DTCs from all nodes 9-04-2009 3-18-00 am.csv）。Excel もしくは他のスプレッドシートエディタにて、この種類のファイルは開けます。

パート５：まとめ

本チュートリアルでは、タスクを自動化するための、Vehicle Spy 上の異なる機能の結合方法を示しました。ファンクションブロックにより、共通のタスクを実行するために簡単なスクリプトを作成できます。グラフィカルパネルでは、ファンクションブロックの制御、またテストの状態や収集したデータの表示が可能です。ファンクションブロックとグラフィカルパネルは、作成者次第で、いくらでも複雑にも単純にもなりますが、重要なのは希望のタスクを実現することです。

[11] 訳注：neoVI FIRE 等の車載ネットワークインターフェース

チュートリアル６：DPS プログラミングの自動化

概要

本チュートリアルでは、ゼネラル・モーターズの DPS アプリケーションを使用して ECU のプログラムを行うため、ファンクションブロックとグラフィカルパネルによる診断ジョブの使い方を学びます。最終的には、2 個のファイルを逐次フラッシュするグラフィカルパネル上のボタンが出来上がります。本チュートリアルはチュートリアル５と同様ですが、使用する診断サービスのみが異なります。完成例は、Logon 画面の Examples タブの、DPS Automation Example を参照して下さい。

本チュートリアルでは以下のステップを通して説明します。

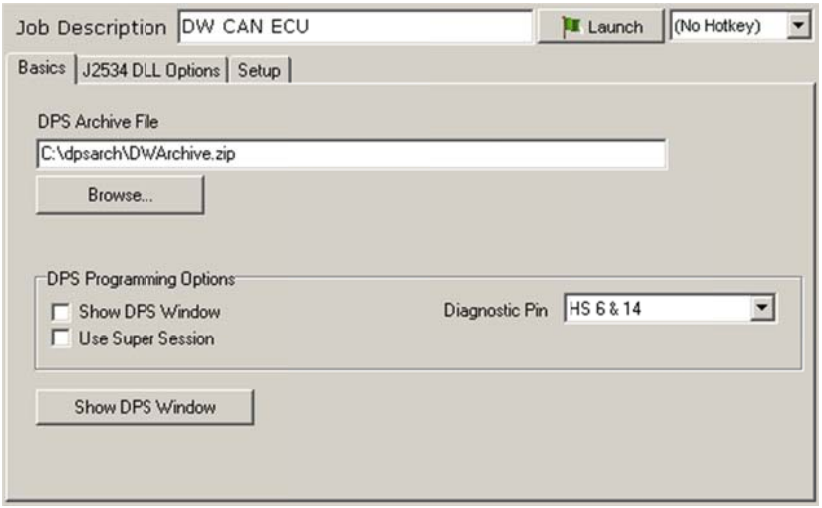
1. 2 つの DPS プログラミングジョブの作成
2. それらのサービスを開始するための、ファンクションブロックの作成
3. エンドユーザのためのグラフィカルパネルの作成
4. 成果物の使用

パート１：DPS プログラミングジョブを作成

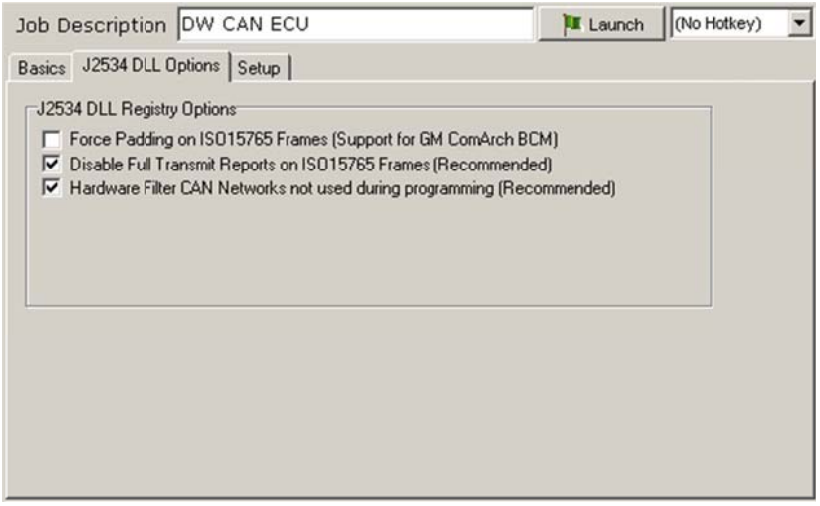
1) **Vehicle Spy の Logon：**Vehicle Spy を開き、Logon Name を選択した後、メニューバーから、File -> New を選択します。

2) **DPS 診断ジョブの作成：**メニューバーから、Vehicle Networks -> Diagnostics Setup を選択し、Setup Diagnostics ウィンドウを開きます。＋ボタンを押し、GMLAN -> DPS Programming を追加してください。

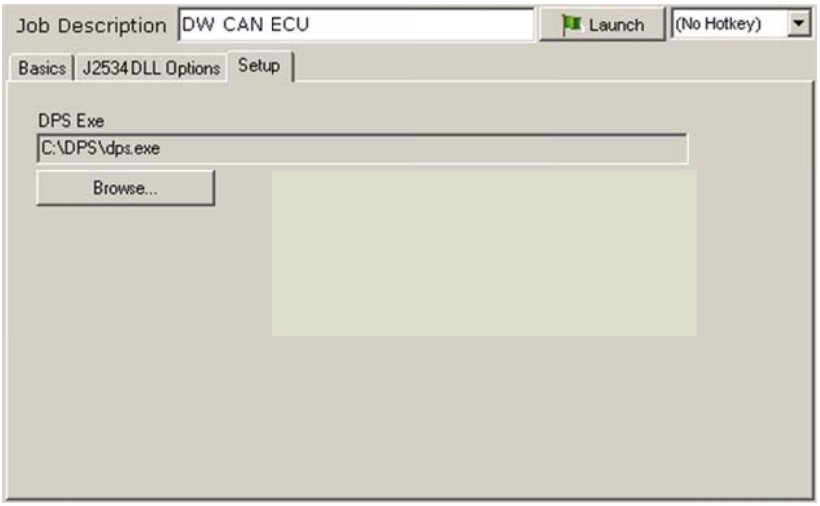
3) **Description と基本設定の編集：**最初に Job Description を変更して下さい。本チュートリアルでは、2 つ以上の DPS プログラミングサービスを設定するので、区別しておく必要があります。Basic タブでは他に、DPS Archive File と Diagnostic Pins の確認が重要です。



4) **J2534 DLL Options の確認：**J2534 DLL Options タブには、J2534 通信を構成するオプションがあります。通常、デフォルトで構いません。



5) **DPS.exe の場所の確認：**Setup タブの DPS Exe 欄では、DPS.exe の場所を指定します。DPS が別のディレクトリにインストールされている場合、Browse ボタンをクリックし、適切なディレクトリを選択して下さい。



6) **他のジョブの作成：**プログラムしたい別の ECU に対して、新たに DPS プログラミングサービスを追加します。完成例(DPS Automation Example.vs3)に従っている場合、適当な DPS アーカイブファイルをそれぞれの診断ジョブに対して適用して下さい。

パート２：ジョブを起動するファンクションブロックを作成

1) **Function Blocks ビューを開く：**次のステップは、作成した診断サービスを制御するためのファンクションブロックの作成です。Scripting and Automation -> Function Blocks から、Function Blocks ウィンドウを開きます。

2) **ファンクションブロックの作成：**＋ボタンをクリック、Script を選択して下さい。ファンクションブロ

ックの記述を、”Program ECUs” のようなわかりやすいものに変更して下さい。

3) **スクリプトの作成：**スクリプト完成にあたり、6 ステップが必要です。以下の表に、必要なステップとその内容を示します。Diag Job Action は、Value 欄をダブルクリックすると、動作させる診断ジョブと動作方法が選択できます。詳細は、Diag Job Action コマンドのヘルプを参照して下さい。

ステップ	コマンド	値	ステップ内容
1	Diag Job Action	Start DW CAN ECU	診断サービスを開始します。
2	Diag Job Action	Wait until Complete DW CAN ECU	選択した診断サービスが終了するまで待ちます。終了後、次のステップに進みます。
3	Wait For	5,000 Sec	5 秒間待ちます。これは必須ではありませんが、スクリプトの動作をユーザが確認できるよう遅延を入れます。
4	Diag Job Action	Start SW CAN ECU	診断サービスを開始します。
5	Diag Job Action	Wait until Complete SW CAN ECU	選択された診断サービスが終了するまで待ちます。終了後、次のステップに進みます。
6	Stop	n/a	このファンクションブロックを終了します。

4) **開始設定を構成：**Start タブで、Manual Start を選択して下さい。これは、次の節で設定するグラフィカルパネルのツールから、ファンクションブロックを開始する上で必要です。

パート３：グラフィカルパネルを作成

グラフィカルパネルを作成し、DPS プログラミングの操作を簡単にしましょう。

1) **グラフィカルパネルを開く：**Measurement -> Panels で、Graphical Panels Editor を開きます。

2) **パネルの作成：**今回、画面の下部の、7 個のツールを用います。

最初は、ファンクションブロックを開始するための Function Block ボタンです。設定すべき重要なプロパティは、以下の通りです。
FunctionBlock: Program ECU
FBlockAction: 0-Start

設定すべき残りのツールを以下に示します。

Text Display
Signal: 最初のジョブに対して、Jobs -> Log -> Value プロパティを選択して下さい（Evaluate as text が選択されていることを確認）。

Text Display
Signal: 2 番目のジョブに対して、Jobs -> Log -> Value プロパティを選択して下さい（Evaluate as text が選択されていることを確認）。

Bargraph
Signals: 最初のジョブに対して、Jobs -> Percent -> Value プロパティを選択して下さい。
Min: 0

LED
Signal: Jobs -> 最初のジョブ名 -> Is Successful プロパティを選択します。

LED

Signal: Jobs -> ２番めのジョブ -> Is Successful プロパティを選択します。

以上により、ファンクションブロックボタンをクリックすると、DPS プログラミングジョブが開始されます。Text Display は、ログからの、現在のテキストを表示します。棒グラフは、プログラミングの進行状況を表示し、LED はプログラミングが完了したときに点灯します。下図に完成例を示します。

パート４：DPS プログラミングパネルを利用

1) **パネルを動作させる：**本チュートリアルを実際に実行するためには、ハードウェアを車両または ECU に接続する必要があります。グラフィカルパネルのボタンをクリックし、作成したファンクションブロックを開始して下さい。DPS 由来のウィンドウが画面にポップアップするかもしれませんが、Vehicle Spy がそれらを管理し、終了時には閉じます。グラフィカルパネルのツールは、進行状況を表示し、プログラミングイベントからの情報を記録するはずです。最初の ECU に関して完了後、2 番目の ECU がプログラミングを開始します。

パート５：まとめ

本チュートリアルでは、1 つのボタンのクリックで、複数の ECU をプログラムする方法を示しました。また、タスクを自動化するために、Vehicle Spy の異なる機能の結合方法を示しました（チュートリアル 5 同様）。

チュートリアル７：診断機能の応用

概要

本チュートリアルでは、より高度な機能の実現のため、複数の診断サービスを協調動作させる方法を説明します。ここで扱うシナリオは下記のとおりです。

- Read DID から Write DID
- ダイナミック DPID で作成された Read DPID

本チュートリアルでは、前チュートリアルが完了していること、及び GMLAN 診断サービスの基礎知識を前提とします。実車またはテストベンチは、有用ですが必須ではありません。重要なのは、本チュートリアルで紹介するコンセプトを理解することです。

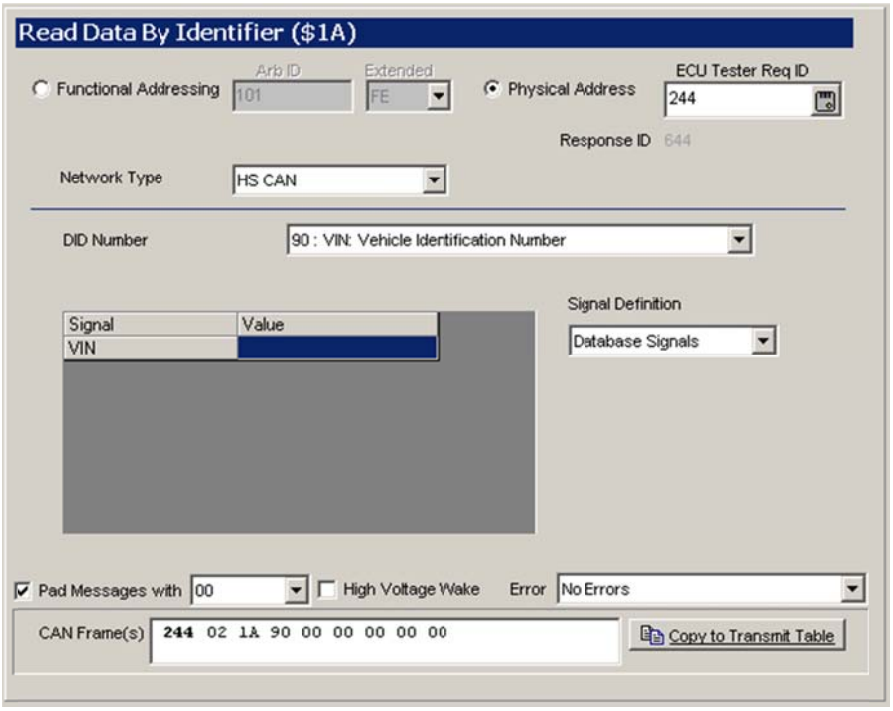
パート１： DID セットアップの読込

この例では、Read DID から取得した DID 情報は、Write DID サービスへと渡されます。これにより、VIN 番号(Vehicle Identification Number)の読込、ECU のプログラミング、そして VIN 番号の ECU への書込などが可能となります。

1) **Vehicle Spy への Logon：**Logon しユーザネームを選択した後、File -> New を選択して下さい。

2) **Read DID ジョブの作成：**次に、DID を読み込むための診断ジョブを作成します。Networks -> Diagnostics として Setup Diagnostics 画面を開き、＋ボタンを押して GMLAN $1A Read DID By Identifier ジョブを選択して下さい。

3) **Read DID ジョブの編集：**次に、Read DID サービスを設定します。もし読書可能な DID にアクセスできる、テストベンチもしくは実車に接続する場合、これを使用して下さい。そうでない場合は、VIN 読込の設定を行います。ECU を Physical Address へとセットし、DID 番号を選んで下さい。VIN としては、DID $90 を選びます。

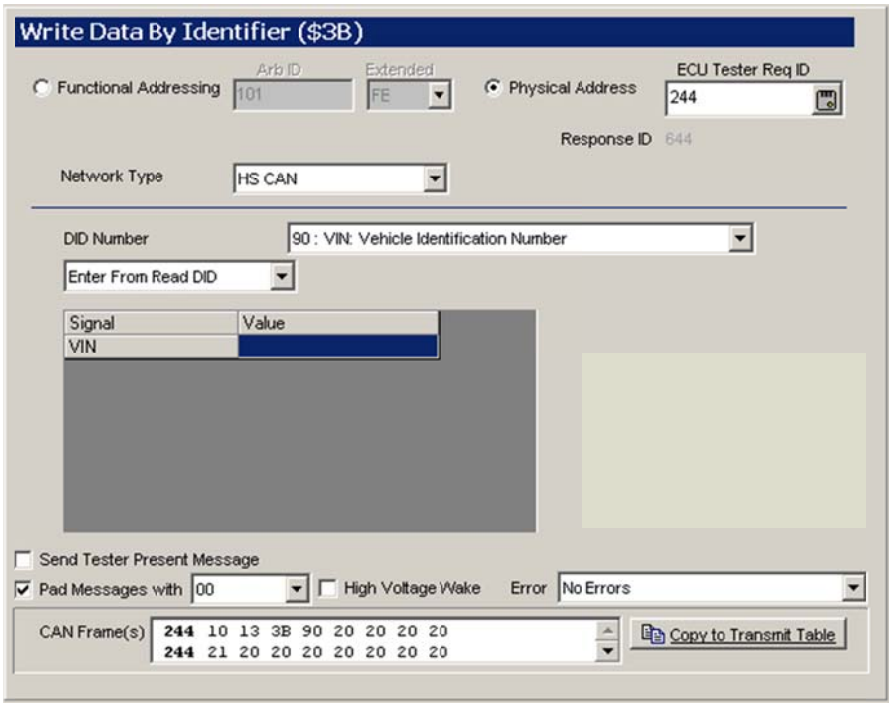


パート２： DID セットアップの書込

1) **Write DID ジョブの作成：** Write DID サービス（$3B）を作成します。＋ボタンを押して、GMLAN $3B Write DID By Identifier ジョブを追加して下さい。

2) **書込する DID の選択：**次に、書込対象の DID を選択します。ここでも、VIN を例として利用します。もし実車もしくはテストベンチを使用する場合、パート 1 で Read DID に対して設定したものと同じ DID を使用して下さい。ECU を Physical Address へとセットし、DID 番号を選んで下さい。VIN としては、DID $90 を選びます。

3) **書込するデータの選択：**最後に、Write DID と Read DID サービス間のリンクを設定します。DID Number の下部にて、書込対象のデータの、入力方法が選択可能です。Enter From Read DID を選択すると、Select Read DID Service と記されたドロップダウンが現れます。ドロップダウン中から、リンクしたい Read DID サービスを選択します。この例では、選択肢は 1 つのみのはずです。



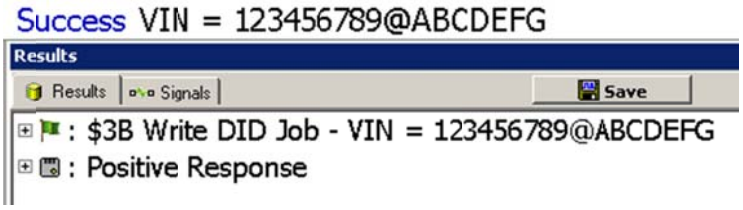
パート３： Read DID の書込

1) **Read DID ジョブの実行：**Diagnostics 画面（Vehicle Networks -> Diagnostics）から、Read DID サービスを実行します。テストベンチまたは実車にハードウェアを接続している場合、リクエストしたデータを受信するはずです。

2) **Write DID ジョブの実行：**ここで、Write DID サービスを実行します。これは、Read DID ジョブで得たデータを用い、ECU に VIN（または他の選択した DID）を書き込みます。

これは、１つの簡単な例に過ぎません。例えば、Write DID サービスは別の ECU に適用することも可能です。別の例としては、Read DID と Write DID 間に、DPS プログラミングを挟むなどが考えられます。

その場合、プログラミング完了時、VIN と他のパラメータを ECU に戻すことが可能です。

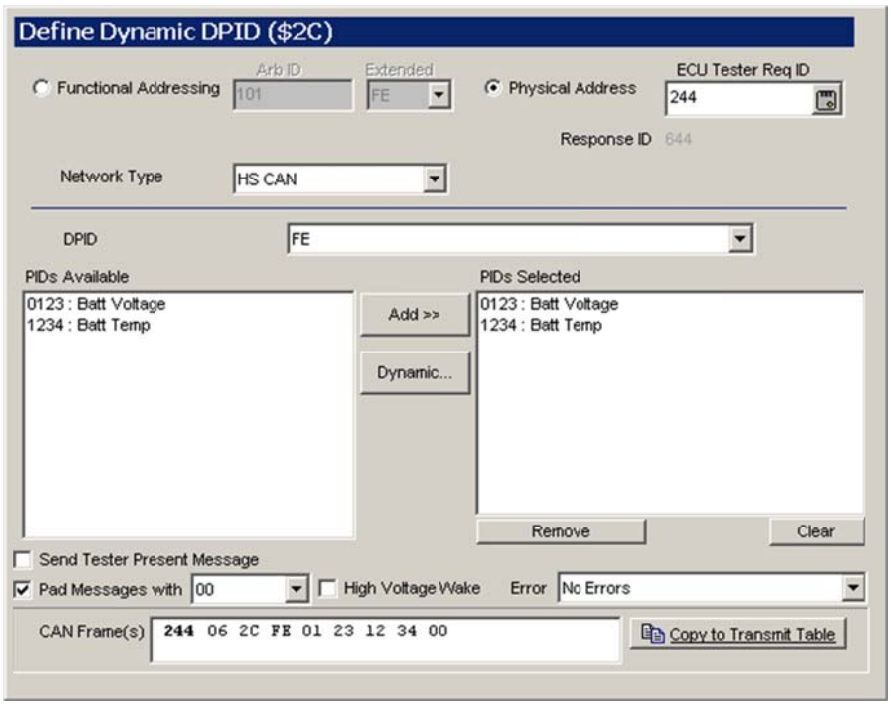


パート４： ダイナミック DPID からの PIDS のリクエスト

Part 4 と 5 では、ダイナミック DPID の使用による、PID の要求を行います。これは、診断サービス Define Dynamic DPID（$2C）と Read DPID（$AA）により実現されます。

1) **PID の作成：**以前と同様、このパートの動作にはハードウェアが必要です。利用できない場合、流れを理解するために単に読み込むのも有効です。最初のステップとして、PID が、データを要求する ECU に対して定義されていることを確認します。この情報は、Vehicle Networks -> ECUs から入力します。ECU を選択、PIDs タブをクリックし、PID データに入力します。

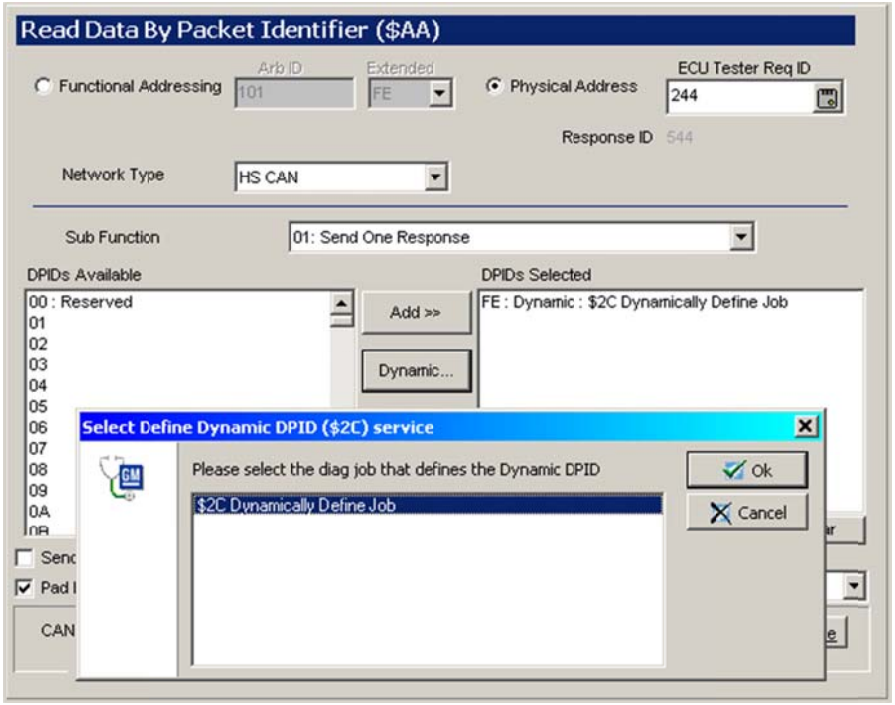
2) **ダイナミック DPID の作成：**次に、PID データを要求するためにダイナミック DPID を作成します。Setup Diagnostics 画面（Vehicle Networks -> Diagnostics）で GMLAN $"2C Dynamically Define Message ジョブを追加して下さい。ECU と Network Type を選択します。PIDs Available リストから PID を選択し、ダイナミック DPID にそれをセットするために Add >> ボタンをクリックして下さい。下図は一例です。



パート５： ダイナミック DPID の要求

1) **ダイナミック DPID リクエストの構成：**最後に、DPID を要求します。このデータは、Read Data By Packet Identifier サービス（$AA）を使用してアクセス可能です。Setup Diagnostics で、Read Data By Packet Identifier サービス（$AA）を追加します。次に、ECU ID とネットワークを設定します。Sub Function は Send One Response の設定を維持して下さい。ダイナミック DPID を Read DPID サービスにリンクするため、Dynamic…ボタンをクリックし、Part 4 からのダイナミック DPID サービスを選択します。例として下図を参照して下さい。

2) **PID を要求：**ここで、PID を要求するために、必要なもの全てを設定します。Diagnostics 画面（Vehicle Networks -> Diagnostics）から、ダイナミック DPID サービスを開始して下さい。ポジティブ・レスポンスを受信した場合、次に Read DID サービスを実行します。PIDs から要求されたデータが表示されるはずです。これは、Vehicle Spy が、ECUs ビューで設定した Diagnostic Database デコーディングを使用して、PID をデコードしたことを意味します。



パート６： まとめ

本チュートリアルでは、複数のタスクを実現するため、複数の診断サービスがいかにリンクされるかを説明しました。ファンクションブロックを用いれば、複数のサービスの送信と、それらの応答の自動化が可能です。

チュートリアル８：CoreMini の使用と設定

概要

本チュートリアルでは、スタンドアローン操作とロギングのための、Coremini の使用方法を説明します。最終的には、データロギングおよび J1979 RPM リクエストを行うスタンドアローン・アプリケーションが得られます。デバイス[12]中の Micro SD カードに保存されたデータの抽出方法も、本チュートリアルに含みます。

このゴールに向けて、下記のステップを実施します。

1. J1979 RPM リクエストの作成
2. データロギングのためのファンクションブロックの作成
3. CoreMini をデバイスにロード
4. データの抽出
5. 収集したデータの表示

パート１：送信メッセージの作成

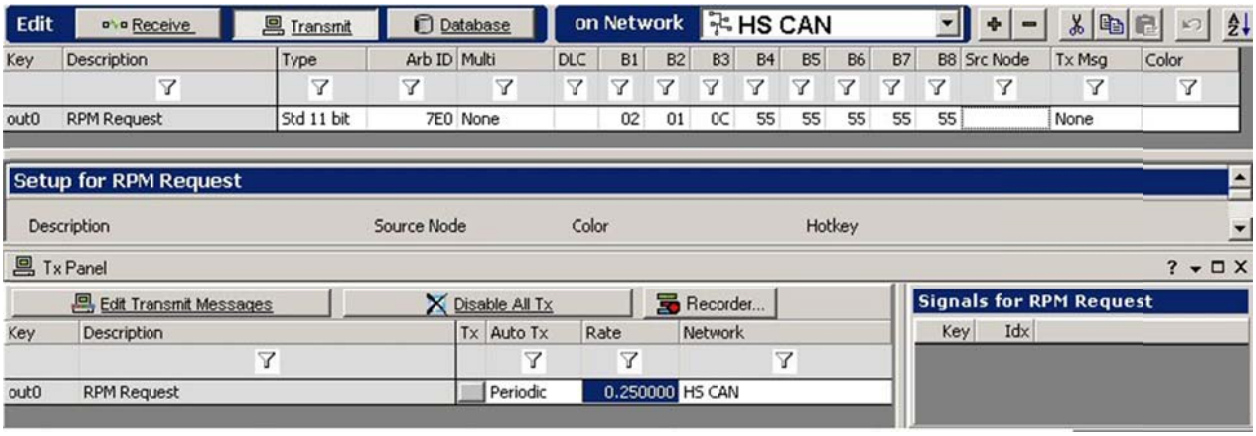
1) **送信メッセージを作成：**Vehicle Spy を開き、Logon 画面にてログインします。次に、以下のパラメータで送信メッセージを作成します（詳細は

[12] 訳注：neoVI FIRE などのデータロガー

2) チュートリアル２：メッセージ送信にて解説済みです）。

Description	RPM リクエスト
ArbID	$7E0
B1	$02
B2	$01
B3	$0C
B4	$55
B5	$55
B6	$55
B7	$55
B8	$55

作成完了の後、送信周期を設定します。Tx Panel を開き、周期を 0.25 秒にセットして下さい（下図）。



上記の設定により、CoreMini スクリプト動作時、このリクエスト・メッセージが 250ms 毎に送信されます。

パート２： データロギング

1) **キャプチャ・ファンクションブロック：**次に、キャプチャ・ファンクションブロックを設定します。この例では、とてもシンプルな設定を行います。

まず Scripting and Automation から Function Blocks を選択します。＋ボタンを押し、Capture を選択して下さい。キャプチャ方法の設定のため、これからいくつかのオプションをセットする必要があり、以下に簡単な例を示します。

Buffer
ここでは、キャプチャしたいメッセージを選択でき、デフォルトでは全メッセージをキャプチャします。この例では、デフォルト設定のままにします。

Start
キャプチャの開始タイミングを設定します。この例では、Start Immediately を選択して下さい。デバイスに電源が投入されると、直ちにメッセージのロギングを開始します。また、Automaticcaly restart when complete も有効にして下さい。このオプションにより、現在のバッファが満たされると、自動的にデバイスは新しいバッファを作成し、ロギングを続行します。これは電源を遮断するまで継続されます。

Stop and Trigger
キャプチャ方法の設定を行います。ここでは簡単のため、Collect in a one-shot buffer を選択し、バッファサイズのメッセージ数に 5000 を入力して下さい(下図)。これにより、デバイスは全メッセージをキャプチャし、それらを 5000 メッセージずつのバッファへと分割します。

Storage
データの保存方法を設定します(下図)。ここではデフォルトのままにします。

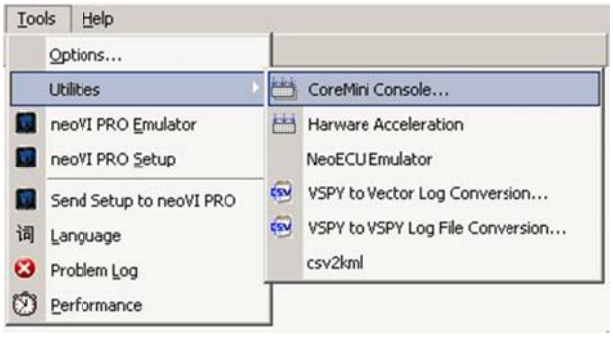
Data
この画面では、ファンクションブロック動作時に、実際に収集されているデータをリアルタイム表示します。これは Vehicle Spy 上でファンクションブロックを実行しているときに限られ、Coremini ロギング時[13]は動作しません。

以上で、キャプチャ実行の上で必要な設定はすべて完了しました。

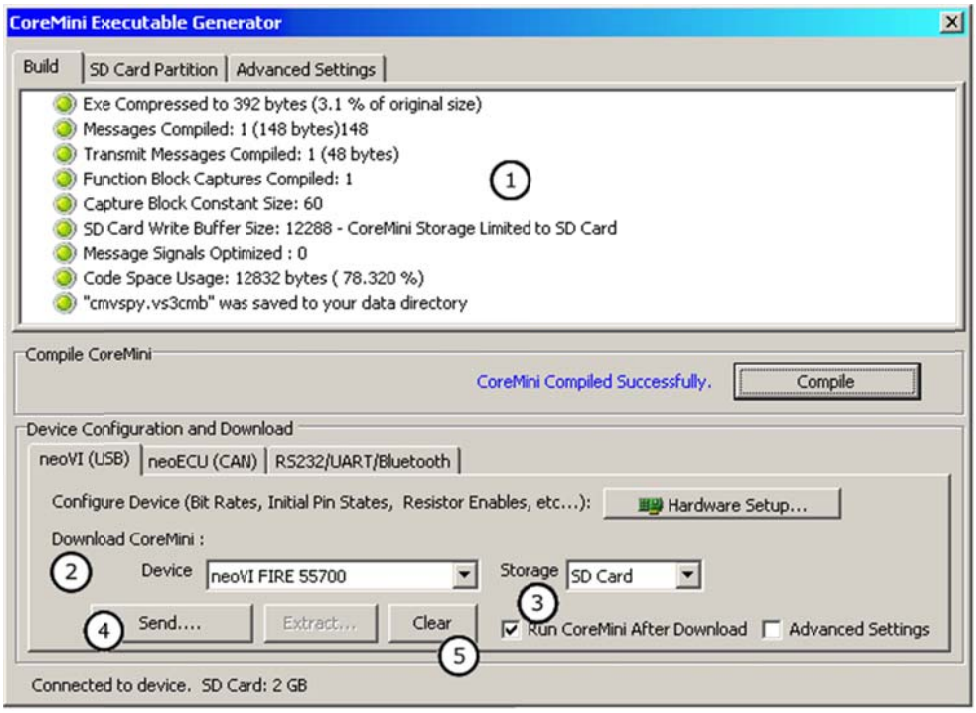
パート３： スクリプトのロード

[13] 訳注：デバイスが PC から独立してロギングを行う、スタンドアローン・モードのこと。

1) **Coremini Console：** Tools -> Utilities から CoreMini Console を開いて下さい。ここで、デバイスに対するスクリプトのロードもしくは削除を行います。



2) **CoreMini Console を使用：**Coremini Console は、デバイスへと転送される各種設定に関して、多数の情報を含みます。下図の①はその出力を示し、エラーや警告はここから確認できます。もしスクリプト内に問題があれば、赤い点で警告表示が発生します。”neoVI (USB)”セクション(下図：②)では、スクリプトのロード先デバイスを指定、Storage セクションでは、さらにそのデバイス内部での保存先を指定します。プレイバックおよびキャプチャスクリプトにおいては、SD Card を選択しなくてはなりません。”Run CoreMini After Download”チェックボックスを有効化すると、スクリプトをロード後、直ちにスクリプトを実行開始します。これを無効化すると、デバイスに電源を再投入後に、スクリプトが実行開始されます。”Send”ボタンで、デバイスへスクリプトを転送します。



3) **メモリからスクリプトを消去：**Clear ボタン(上図⑤)押下により、スクリプトをデバイスから消去できます。消去後、スクリプトを再実行する場合は、改めてスクリプトをデバイス上に転送する必要があります。

パート４：スクリプトの実行

1) **CoreMini を実行：**”Run CoreMini After Download”が有効化されている場合、Send ボタン押下後、直ちにスクリプトの実行が始まります。その時、赤い LED が点滅し、スクリプトが実行中であることがわかります（スクリプトにより LED の挙動が変更されている場合を除く）。

2) **スクリプトを制御：**以下の表の通り、デバイスへの電源の投入の仕方により、スクリプトの実行状況が変わります。

接続順序によるデバイスモードと LED パターンの変化

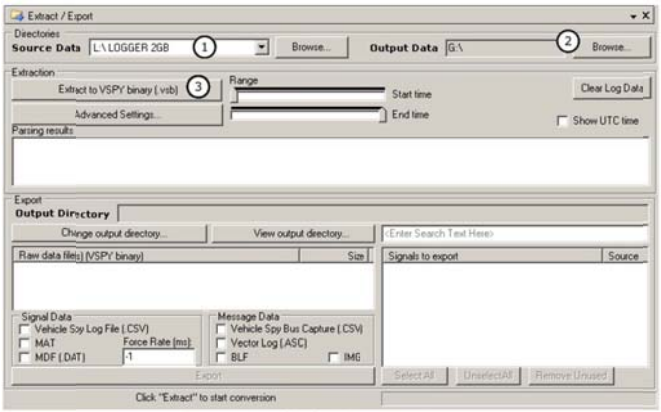
接続順序	デバイスモード	既定の LED 動作
電源のみ接続	CoreMini スタンドアローン・モード。スクリプトがロード済みならば、デバイスはスクリプトを実行します。	赤が点滅
USB 接続 -> 電源接続	PC モード。CoreMini は動作しません。	緑が点滅
電源接続 -> USB 接続	CoreMini スタンドアローン・モード。スクリプトがロード済みならば、デバイスはスクリプトを実行します。 ただし、PC と接続してバスをモニターするなどの利用方法が可能です。	赤と緑が交互に点滅

3) **データを取得：**デバイスを車載ネットワークに接続し、データ収集のためのスクリプトを走らせて下さい。

パート５：データの抽出

1) **Micro SD カード：**Vehicle Spy は、Micro SD カードからデータを抽出し、PC 上にコピーする”Extract”ツールを標準で搭載しています。まず、デバイスから Micro SD カードを抜き取り、それを Micro SD カードリーダに差し込んで下さい。そのリーダを PC の USB ポートに差し込むと、Windows にそれがディスクドライブとして認識されますので、そのドライブ名を確認して下さい。

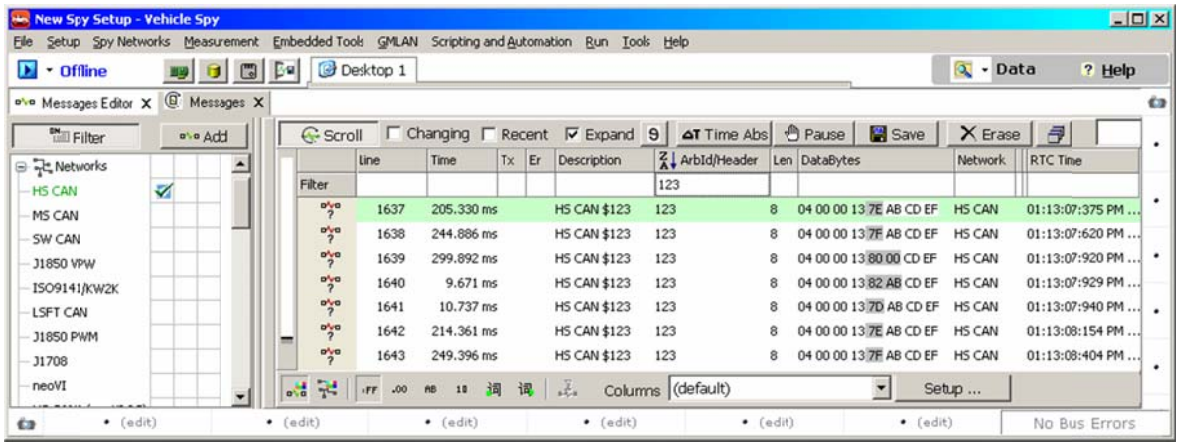
2) **SD カード抽出：**Tools -> Utilities -> Extract / Export と開いて下さい（下図）。



初めに、抽出対象の Source Data ドライブを指定します。ドライブ名は、前項で確認したものと同じで、かつ LOGGER というラベル及びサイズ情報を有しているはずです。次に、Output Data の項の Browse ボタンを押して、データの抽出先を指定します(上図②)。最後に、”Extract to VSPY binary(.vsb)”ボタン(上図③)を押すと、SD カードからデータ抽出が始まり、所定のディレクトリへとデータが出力されます。完了後、Windows のエクスプローラが、自動的に出力先のディレクトリを開きます。

パート６：データの表示

1) **バッファを表示：**データの抽出が完了したので、その内容を見てみましょう。Vehicle Spy には、抽出したデータを表示する際、２つの方法があります。１つは、抽出されたデータ(.vsb)を、Message ビューへとドラッグ&ドロップする方法、もう一つは、File -> Review Buffer から開く方法です。どちらの方法でも、キャプチャしたデータが、Message ビュー上に表示されます。もしデータベースを設定済みであれば、それらのデータは、デコードされた状態で表示されます。このとき、Vehicle Spy のフィルタ機能を利用すると、興味のあるメッセージのみを表示する上で有用です。
この例では、RPM リクエストが表示されますが、もし HS CAN 上での J1979 をサポートする車両に接続してロギングした場合、RPM データを伴うレスポンスを受信しているはずです。



概要

データのキャプチャのため、下記の手順を実行しました。

1. J1979 RPM リクエストの作成
2. データを保存するファンクションブロックの作成
3. CoreMini をデバイスにロード
4. データの抽出
5. 収集したデータの表示

これはごく単純な例であって、CoreMini はこれよりも遥かに広い応用範囲を有します。例えば、簡単なスクリプト・ファンクションブロックにより、ECU の機能をシミュレートしたり、また診断データをリクエストすることが可能です。またデータのキャプチャは、特定の条件や、入力信号に応じた設定が可能です。

チュートリアル９：C Code Interface 概要

概要

本チュートリアルでは、C Code Interface の基礎を説明します。C Code Interface の利用には、Visual C++ Express 2005 もしくはそれ以上のバージョンが必要です。Express 版は、無料でマイクロソフトの HP よりダウンロード可能です。ここでは、C 言語および Visual Studio の基礎的知識を前提としてチュートリアルを進めます(Visual C++ Express 2008 を利用)。

Vehicle Spy で使用する部分

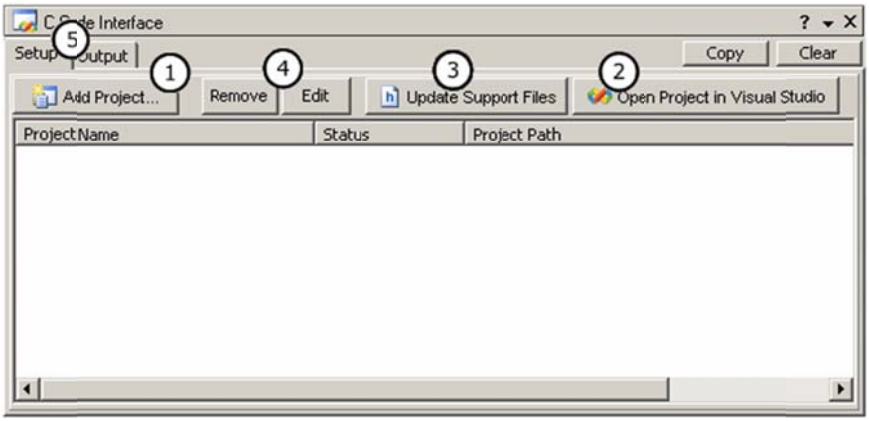
- C Code Interface
- Message Editor
- Application signals

必要なハードウェア

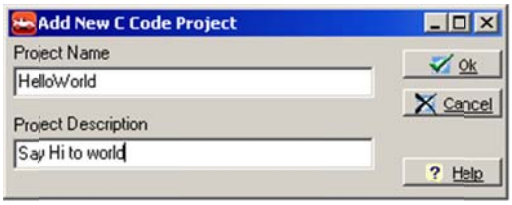
シミュレーションモード, neoVI FIRE, neoVI RED, neoVI Yellow, ValueCAN3 のいずれか

パート１：プロジェクト作成

1) **Vehicle Spy を開く：**まず、通常通り、Vehicle Spy を起動し Logon します。そして、Scripting and Automation -> C Code Interface を開きます(下図)。



2) **プロジェクトを追加：**Add Project ボタン(上図①)、続けて"New Project"を選択することで、新しいプロジェクトを追加します。プロジェクト名と、簡単な説明文を加えて下さい(下図)。OK を押すと、Visual Studio がその新しいプロジェクトと共に開くはずです。



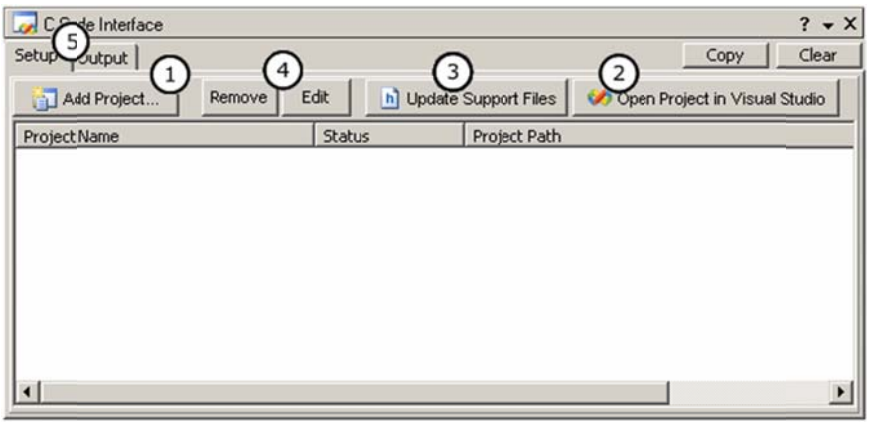
3) **ビルドと実行：**Visual Studio 上にて、プロジェクトをビルドし、Vehicle Spy のスタートボタンを押して

下さい。Output タブ内にて、”Another Second happened”という出力が一秒ごとに現れるはずです。この表示は、先の C コードから出力されています。Visual Studio 上にて、メインのプロジェクトファイルは SpyCCode.c であり、Spy_Main 関数がこの文字列を生成しています。試しに、文字列を”Hello World of Vspy!”と変更してみましょう (Vehicle Spy を停止し、改めてビルドすることを忘れずに) 。

パート２：イベントの追加

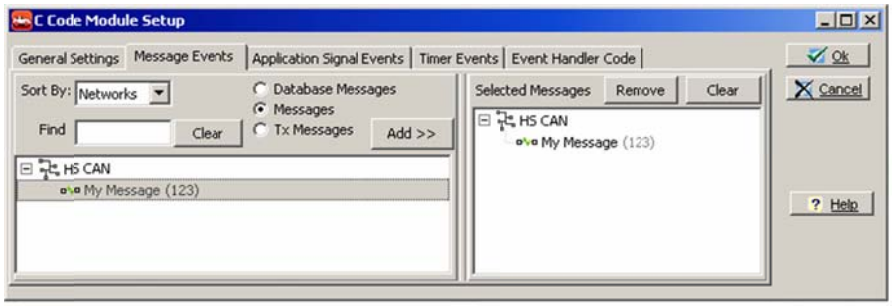
1) **Vehicle Spy 設定：**本チュートリアルでは、２つのイベントを追加します。１つ目はメッセージ受信時、２つ目はアプリケーションシグナルが変化した時、です。”My Message”という名の ID 123 の受信メッセージ、およびアプリケーションシグナル”My App Signal”を作成して下さい。

2) **C Code Interface：**Setup タブにて、自身のプロジェクトを選択し、”Update Support Files”をクリックして下さい (下図③) 。これにより、Vehicle Spy 上の新たなデータ項目[14]が、ヘッダファイルへと追加されます。もし Visual Studio が開いていれば、ファイルの変更が行われたことを示す警告が現れるでしょう。その場合、”Yes to All”をクリックして下さい。



3) **イベントを追加：**プロジェクトが選択された状態で、Edit ボタンをクリックして下さい(上図④)。Message Events 内にて My Message を、Application Signal Events にて My App Signal を追加します (下図) 。次に、Event Handler Code にて Copy をクリックして下さい。これがイベント用のコードになります。このコードを Visual Studio の SpyCCode.c 内に、ペーストして下さい。

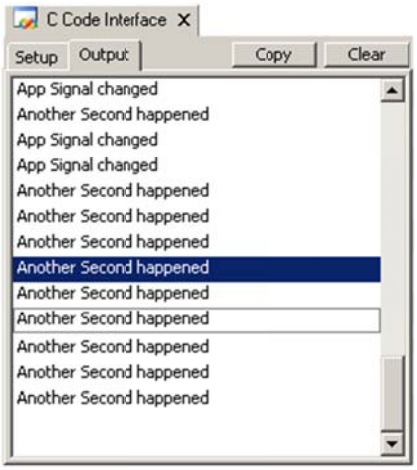
Spy_MG_My_Message_HS_CAN イベントに対して、printf(“There’s my Message!!!\n”); を追加し、SpyAppSign_AS_My_App_Signal に対して、printf(“App Signal changed\n”); を追加してみましょう。



[14] 訳注：新規に定義した受信メッセージ及びアプリケーションシグナルの内容を、C コード側に反映させる上で必要な操作

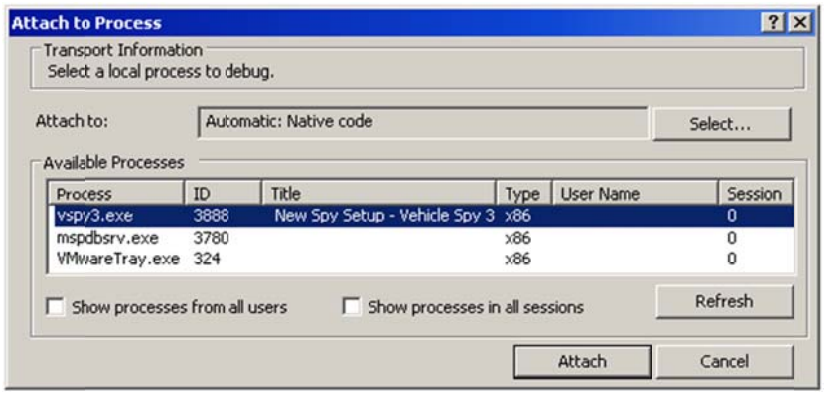
パート３：サンプルの実行

1) **サンプルの実行：**C コードをビルドし、Vehicle Spy のスタートボタンを押して下さい。Output 画面に、イベント発生時、各テキストが出力されるはずです (下図) 。次に、グラフィカルパネルを用いて、簡単にアプリケーションシグナルを変更してみましょう。



パート４：デバッグ

1) **デバッグ：**Visual Studio のデバッグ機能を利用した、デバッグが可能です。プロジェクトをリビルドした後、Visual Studio 上にて"ツール -> プロセスにアタッチ"を選択し、vspy3.exe に対してアタッチボタンを押下します(注：Visual C++ 2010 Express では、"ツール -> 設定 -> 上級者用の設定"を有効化しないと、"プロセスにアタッチ"項目が表示されません)。



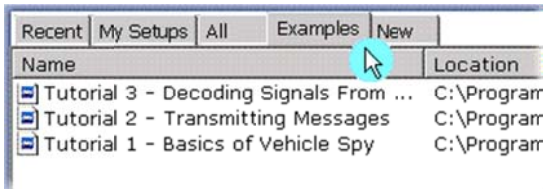
ブレークポイントを適当に設定し、Vehicle Spy のスタートボタンを押してみましょう。Vehicle Spy における C コードのデバッグは、一般的な他の C プロジェクトと同様に行うことが可能です。

パート５： まとめ

1) **結果：** Vehicle Spy における C コードの作成、編集、デバッグの基礎を学びました。

3.3. チュートリアルまとめ

お疲れ様でした！以上で、全チュートリアルが完了です。
これで、Vehicle Spy を扱う上での基礎は十分に得られました。より複雑な機能が必要な場合は、以降のヘルプを参照して下さい。
Intrepid Control Systems, Inc.は、定期的に本チュートリアルの拡充を行なっています。追加のトピックをご要望の場合は、どうぞご連絡下さい。

3.4. チュートリアル用ファイル

下図の通り、チュートリアル用のファイルは、Logon 画面の Examples タブからアクセスが可能です。