

CANerator CANopen

ソース コードの生成およびコンフィギュレーション

CANerator CANopenは、ベクター社のCANopenデバイス用組込ソース コードのコンフィギュレーション カスタマイズ情報を管理し、オブジェクト ディクショナリを自動的に作成するためのツールです。

特徴/優位性

CANopen通信プロファイルは、組込ネットワークでの使用において、その高い拡張性で定評があります。ベクター社のCANopen組込ソース コードは、CANopen通信に必要な機能を備えており、また同時にリソースの小さなマイクロ コントローラに対応可能な機能も備えています。その為、ベクター社のCANopen Masterソース コードおよびCANopen Slaveソース コードは、用途によって様々なコンフィギュレーションできるように設定されています。

通常、こうした柔軟性を実現するには、開発プロセスは複雑になります。さらに、開発されるさまざまなデバイスのすべてに対応するために、ソフトウェアの再利用性やより柔軟な適応性が求められています。

また、ソース コードを実際にデバイス仕様に合わせてコンフィギュレーションする以外に、CANopenデバイスの開発元は、オブジェクト ディクショナリも作成する必要があります。このオブジェクト ディクショナリは、アプリケーションとCANopen通信間のインターフェイスを形成し、またデバイスの特性を記述する役割を果たします。

CANerator CANopenを使用することにより、ソフトウェア開発元は、前途の様な開発負担を減らし、開発プロセスにかかる時間を大幅に短縮化することができます。

機能

CANerator CANopenには、以下の機能があります。

- > #define定義のユーザー フレンドリなインターフェイスによるソース コードのコンフィギュレーション
- > オブジェクト ディクショナリの作成とソース コードの自動生成
- > EDSファイルの自動作成
- > 簡単なプロジェクト管理

これらの機能について、以下で詳しく説明いたします。

応用分野

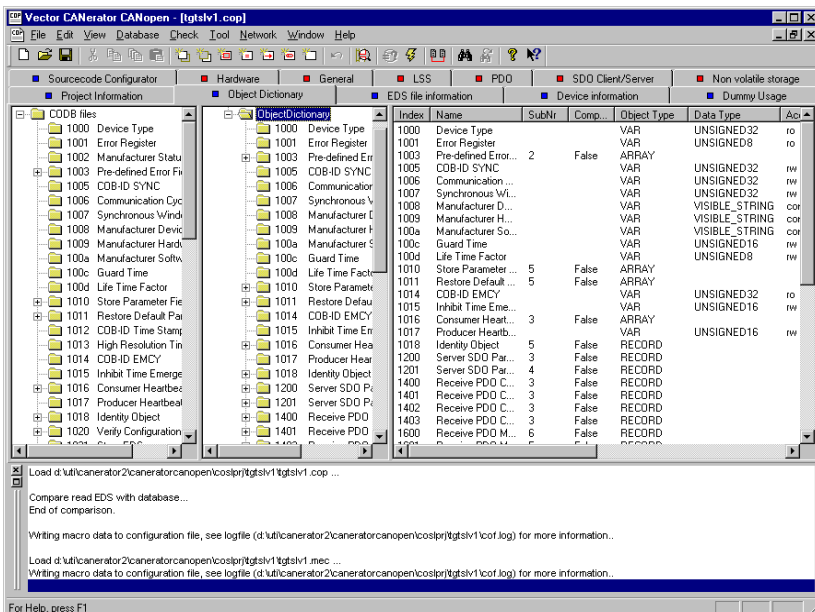
CANerator CANopenは、ベクター社のCANopen Masterソース コードおよびCANopen Slaveソース コードとともに使用するためのものです。

ソース コード コンフィギュレータ

CANerator CANopenは、ソース コード (#definesマクロ定義)に関連するすべての設定をツリー ビューで表示します。

別にインストールされたCANopenソース コードの設定をチェックし、#defineマクロ定義に関連するグループごとに表示します。

さらに、各設定にはコメントと値を割り振れる範囲も表示されます。よって、型に依存した認証によりさまざまなマスクに異なるデータ型が割り当てられます。



CANedsによるEDSファイルの作成

オブジェクト ディクショナリの生成

CANeds (EDSエディタ)が完全に統合されているため、CANedsツールのすべての機能を使用することができます。CANedsについては、CANeds製品情報を参照してください。CANedsにより、オブジェクト ディクショナリのすべての標準オブジェクトをドラッグ&ドロップで作成することができます。さらに、メーカー固有のオブジェクトを定義することも可能です。オブジェクトには、必要に応じて変更可能なデフォルト値が与えられています。

開発元がオブジェクトを指定すると、CANerator CANopenは、階層ツリー構造でオブジェクト ディクショナリを表示します。次に、ソース コード中の変数がオブジェクトに割り当てられ、これらは、最終的にはソース コードのプロトコル固有部分とアプリケーション間のインターフェイスとなります。また変数名のルールを利用し、CANハードウェアの各ポートを直接割り当てることもサポートしています。オブジェクト ディクショナリが作成されると、ソース コードの対応する部分が自動的に作成されます。

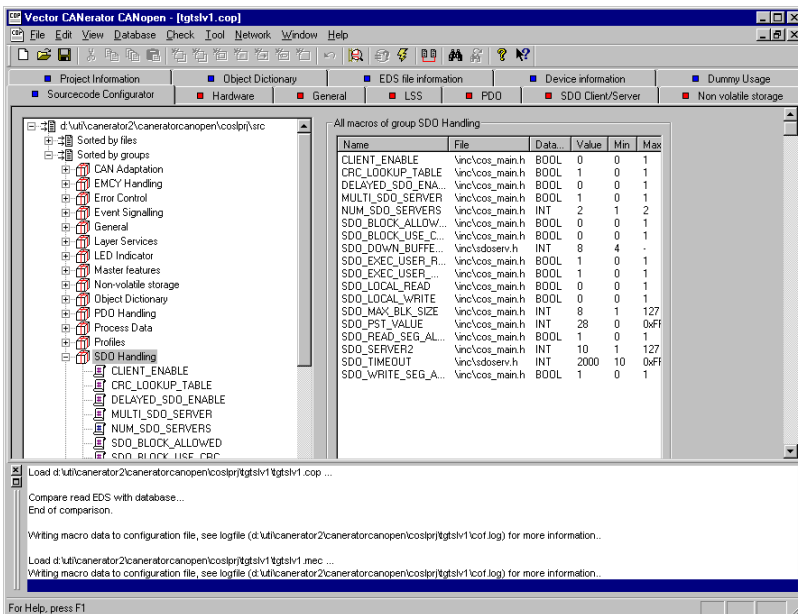
Electronic Data Sheet (EDS)

Electronic Data Sheetが、同時に生成されます。これは、コンフォーマンス テストと同じレベルでテストすることができます。

プロジェクト管理

CANerator CANopenにより、デバイス設計のためのプロファイルを手軽かつユーザーフレンドリに管理することができます。プロファイルは、一連の設定が作成されると、独自の名前で保存できます。このタイプのさまざまなプロファイルを、別デバイス開発や同じプロジェクトで使用することができます。たとえばプロジェクトにデバッグ情報を提供することもしないこともでき、また簡単な操作で複数のプロファイル間を切り替えることができます。

また、他の応用として、ソースコードを基本としたさまざまなデバイスタイプのためのプロファイルを作成することもできます。これにより、シングルソース方式をより簡単に実現できます。これらのプロジェクトを文書化として、HTMLページが生成され、この中に一般的なプロジェクト情報の他に、すべてのオブジェクトとその属性のリストとすべての設定内容が含まれています。



ソースコードコンフィギュレータ:
すべての#define設定をユーザーフレンド
リな方法で編集可能