

CANopen Slave ソースコード

CANopenのための組込ソフトウェア

CANopen Slaveソースコードは、CANopenスレーブデバイスに必要な機能の実装に利用できます。また、デバイス単体ではなく、装置に内蔵されるスレーブノードのための通信機能としても利用できます。

特徴/優位性

CANopen Slaveソースコードを使用することで、製品開発にかかる時間を大幅に削減できます。CANopenプロトコルとのインターフェイスはシンプルにできているため、ユーザーは各自のアプリケーション開発に専念できます。ソースコードは、CANopen DS-301通信プロファイルの最新バージョンに対応しているため、CANopenに準拠したプロジェクトの実装を確実にこなせます。

内部アプリケーションをオブジェクトディクショナリーへ簡単に関連付けるための基本エントリーが用意されています。また、これら既存エントリーの実装方法に沿うことで、独自エントリーの作成も容易です。

機能

CANopenソースコードには、以下のような機能があります。

- > SDOサーバー
- > PDO/SYNCハンドラー
- > NMTスレーブ (Life Guardingをサポート)
- > ユーザー仕様に合わせた柔軟なカスタマイズ
- > 様々なハードウェアプラットフォームへの対応実績
- > 開発支援ツール: CANerator CANopenを利用したソースコードのカスタマイズおよび、EDSからオブジェクトディクショナリーの自動生成

Slaveソースコード製品のオプションとして、次の拡張パッケージを用意。ユーザー仕様に合わせて追加が可能です。

- > Mini-Masterオプション
- > LSS Master
- > LSS Slave

応用分野

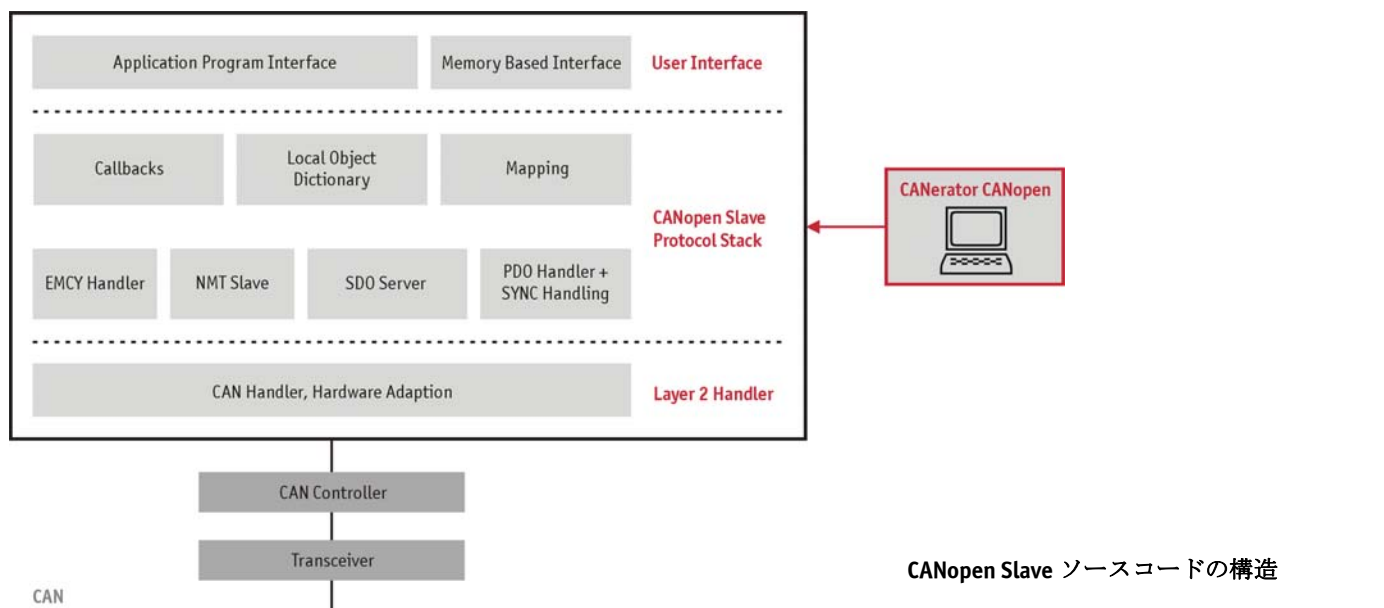
ソースコードは、ユーザーがCANopen機能を実装する場合に利用いただけるものです。特にマイコンへの組込実装を意識して、使用リソースを少なくするように考慮されています。

SDO サーバー

SDOサーバーは、オブジェクトディクショナリーの読み書きを行います。最大2つのSDOコネクションを並行して実装可能で、Expedited転送、Segmented転送、Block転送のすべてのSDOプロトコルをサポートしています。

PDO ハンドラー

PDOの初期設定は、ユーザーが独自の環境に合わせてある決まった手順でコンフィギュレーションされます。ソースコードは、1バイトを最小単位としたマッピングパラメーターのコンフィギュレーションをサポートしています。デバイスが送受信するすべてのPDOを、そのデバイスが持つオブジェクトディクショナリーに登録します。PDOの受信をイベントとして、ユーザーはコールバック関数を呼び出すようにソースコードをカスタマイズでき、この関数内に独自の処理を追加できます。ユーザーは最大200個のPDOを定義でき、ソースコードは、それらの定義に従ってデータ変更によるイベント、またはSYNCメッセージの受信に同期もしくは周期送信イベントにより、アプリケーションの介在なくPDOをハンドリングします。



CANopen Slave ソースコードの構造

ベクターのCANopen Slaveソースコードは、多くの市販マイクロコントローラーに対応した実績があります。最新の情報は、下記URLをご覧ください。

http://www.canopen-solutions.com/canopen_embedded_jp,,331190.html

NMTスレーブ

ソースコードは、Error Control サービスの Guarding と Heartbeatの両方の機能をサポートしています。他のノードから送信されたHeartbeatをモニタリングする機能もあります。(Heartbeatコンシューマー) この機能を利用することで、ネットワークに非常に有効な監視機能を持たせることもできます。また、ユーザー固有のEmergencyメッセージを送信することも可能です。

ソースコードは、パラメーターの不揮発メモリーへのセーブ/ロードもサポートしています。この機能はセーブ/ロードをファイルシステムのように操作できるようにしています。

Mini-Master オプション

多くの組込ネットワークで、すべてのCANopenマネージャー機能を必要としているわけではありません。多くの場合、機能を制限することによって、リソースの使用を最小限にし、また簡単に使用できることに重点を置くことがあります。

Mini-Masterオプションは、このような目的のために用意しています。このオプションには、以下の機能があります。

- > NMTマスター: システム起動時のNMTスレーブデバイスの状態遷移の制御。Node-Guardingによるスレーブデバイスの監視、Heartbeatによる相手デバイスの監視機能は、Slaveソースコードに標準でインプリメント済です。
- > SDOクライアント: スレーブデバイスは、SDO転送を自分から始めることもあります。これにより、他のノードのオブジェクトディクショナリー内の設定を調整することができ、PDOでは使用できないデータ(例: 診断情報)を読み出すこともできます。

LSS (Layer Setting Services) オプション

CANopenデバイスは、まず最初にボーレートとノード番号を設定する必要があります。スイッチによる設定ができないデバイス(防水型のセンサーなど)は、CANバス経由でこれらの値を設定します。このため、CANopenではLSSサービスを定義しています。LSSオプションパッケージにより、スレーブに必要なLSS機能をソースコードモジュールとして提供しています。

ソースコードの実装

ソースコードをコンパイルする前に、デバイス仕様に合わせて機能を限定することで、非常に限られたリソースのシステムでもCANopenプロトコルを実現することができます。ソースコードの細かなカスタマイズは、インクルードファイルに記述されている#defineマクロを利用しています。設定ツール: CANerator CANopen(ソースコードとは別製品になります)を使うと、このカスタマイズをより簡単に行うことができます。

ローカルオブジェクトディクショナリー

CANopen Slaveソースコードのオブジェクトディクショナリーは、ユーザーが簡単に拡張できるように考えられています。これは、追加するパラメーターエントリーをあるC言語ファイルに追記して行います。オブジェクトディクショナリーには、各オブジェクトの属性情報と実メモリー上のパラメーターのポインターを記述します。各オブジェクトに対して、SDOリクエストがあった際にユーザーが定義した関数を呼び出すように実装することもできます。これにより、例えば、ダウンロード対象オブジェクトに書き込める値であるかの判断処理を加えるようにもできます。CANerator CANopenは、オブジェクトディクショナリーのコードを自動生成する機能も持っています。その他の機能については、CANerator CANopen製品の情報を参照ください。

指定ハードウェアプラットフォームへの対応

Slaveソースコードをマイクロコントローラー上に実装するには、その対象マイクロコントローラーを用意する必要があります。どのコンパイラーを利用するのも重要で、ご希望のマイクロコントローラーを指定していただけます。ベクターで過去に出荷実績のある組合せをリストとして用意していますのでご参照ください。

開発プロジェクトのサポート

ベクターでは、CANopen Slaveソースコードをお客様の環境へ実装する際のサポートサービスを用意しています。サポートサービスは、実装だけでなく、要求に合わせたトレーニング、プロジェクトの開発支援(レビュー、シミュレーション環境構築)、ワークショップなどとして提供しています。