

RTC-CANopen リファレンスマニュアル ～RTC-CANopen の構成方法～

1.1 版

RTC-CANopen 0.4.1 対応

2011 年 6 月 7 日

芝浦工業大学

ヒューマンロボットインタラクション研究室

版数	改版日	改版内容
1.0	2011/04/12	初版として発行
1.1	2011/06/07	本書の対象者の項を追加

目次

1.	はじめに.....	4
1.1.	本書の対象者.....	4
1.2.	関連文書.....	4
2.	RTC-CANopen の導入方法.....	5
2.1.	動作環境.....	5
2.2.	RTC-CANopen を利用する上で必要なもの	5
2.3.	Eclipse への RTC-CANopen のツールの導入	35
2.4.	CAN ネットワークインターフェイスの導入.....	6
2.5.	DCF の作成(Maxon Motor 社の EPOS の場合).....	10
2.6.	RTC-CANopen Builder のよる DCF の変換及び XML の作成	35
2.7.	RTC-CANopen System Editor による RTS の作成	40
2.	ライセンス.....	46
	参考文献	46

1. はじめに

本書は RTC-CANopen の導入方法及び構成方法について説明する。

1.1. 本書の対象者

本書は RT ミドルウェア, 及び RTC についての基本知識を有する利用者を対象としている。RT ミドルウェア, RTC については下記を参照のこと。

OpenRTM-aist Official Website:

<http://www.openrtm.org/openrtm/ja/>

リファレンスロボットを利用するためには RTC-CANopen の実行ファイル (RTC-CANopen Manager, PnP Manager, Status Manager 及び ProxyRTC) と RTC-CANopen のツール(RTC-CANopen Builder 及び RTC-CANopen System Editor)が必要となりますが、これらはライセンスの都合上配布を行うことはできません。利用したい方は下記のアドレスにご相談ください。

Mail : shibaura.hri.goiken@gmail.com

1.2. 関連文書

本書に関係する文書を表 1 に示す。特別に版を指定するものを除き、最新版を使用するものとする。

表 1 関連文書

文章名	文章内容
RTC-CANopen リファレンスマニュアル ～RTC-CANopen 利用例～	リファレンスロボットのハードウェア及びソフトウェアの構成の仕方及び RTC-CANopen の起動の仕方が記載してある
RTC-CANopen リファレンスマニュアル ～RTC-CANopen 詳細説明～	RTC-CANopen の機能及びシステム構成やリファレンスロボットに使用しているコンポーネントの詳細な説明が記載してある。

2. RTC-CANopen の導入方法

2.1. 動作環境

RTC-CANopen は PC にて動作する． RTC-CANopen の動作環境を以下に挙げる．

対応 OS : Microsoft Windows 7

Microsoft Windows XP

CPU : Windows Intel Core2Duo 1.6GHz で動作確認

メモリ : 2GB で動作確認

インターフェイス : PC カードスロット

CAN ネットワークインターフェイス : CANcardXL(Windows)

OpenRTM-aist : OpenRTM-aist-1.0.0-RELEASE^[1]

RTC-CANopen : RTC-CANopen 0.4.1

2.2. RTC-CANopen を利用する上で必要なもの

RTC-CANopen を導入する上で必要なものを以下に挙げる．

- RT-Middleware の動作する実行環境(OpenRTM1.0.0RELEASE の C++版)
- CANopen 製品(必要に応じて DCF を作成するツール)
- ベクター・ジャパン株式会社製 CANcardXL
- PC

なお, RTC-CANopen はベクター・ジャパン^[2]のソースを利用しているため, ベクター・ジャパン株式会社製 CANcardXL などのベクターのソースを利用した CANopen 製品で動作する．(現在, 確認済みは CANcardXL および CANcaseXL)

しかし, 現在 RTC-CANopen をオープンソース化する動きがあり, 今後オープンな形で RTC-CANopen を提供する予定である．

2.3. CAN ネットワークインターフェイスの導入

まず、PC と CANopen 製品の通信を行うためのデバイスであるベクター・ジャパン株式会社製 CANcardXL のドライバのダウンロードを行う。CANcardXL のドライバはベクター・ジャパンのホームページ http://www.vector.com/vj_index_jp.html にある。



図 1 CANcardXL

ベクターホームページの項の「ダウンロード」をクリックする。

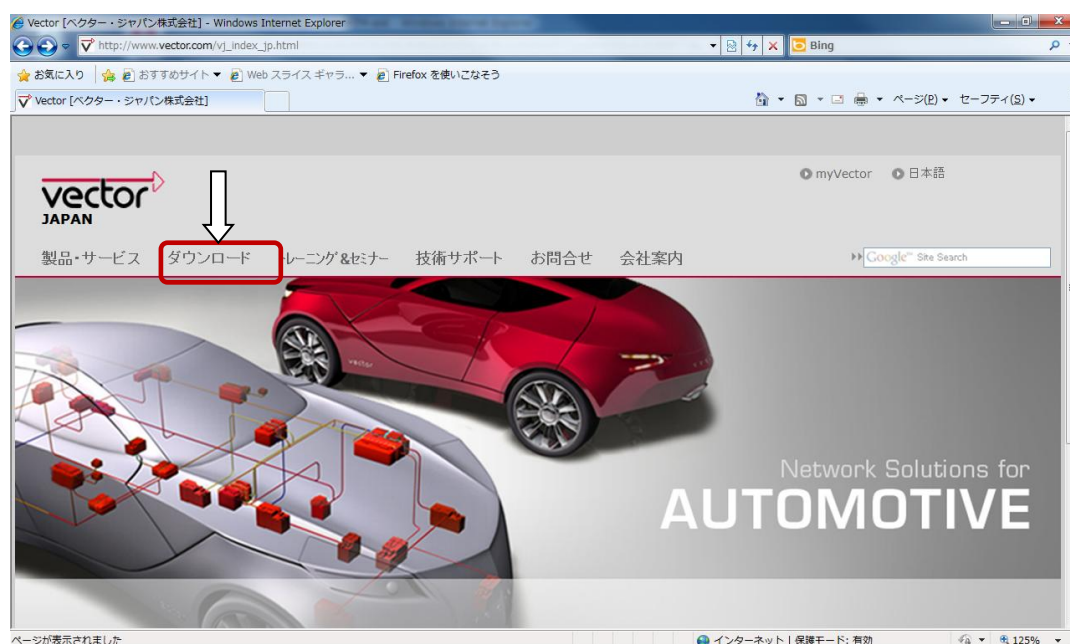


図 2 Vector Japan Web サイト

次に製品の欄を CANcardXL, 種類をドライバに, バス/プロトコルを CANopen に選択し, 「選択結果を見る」をクリックする。

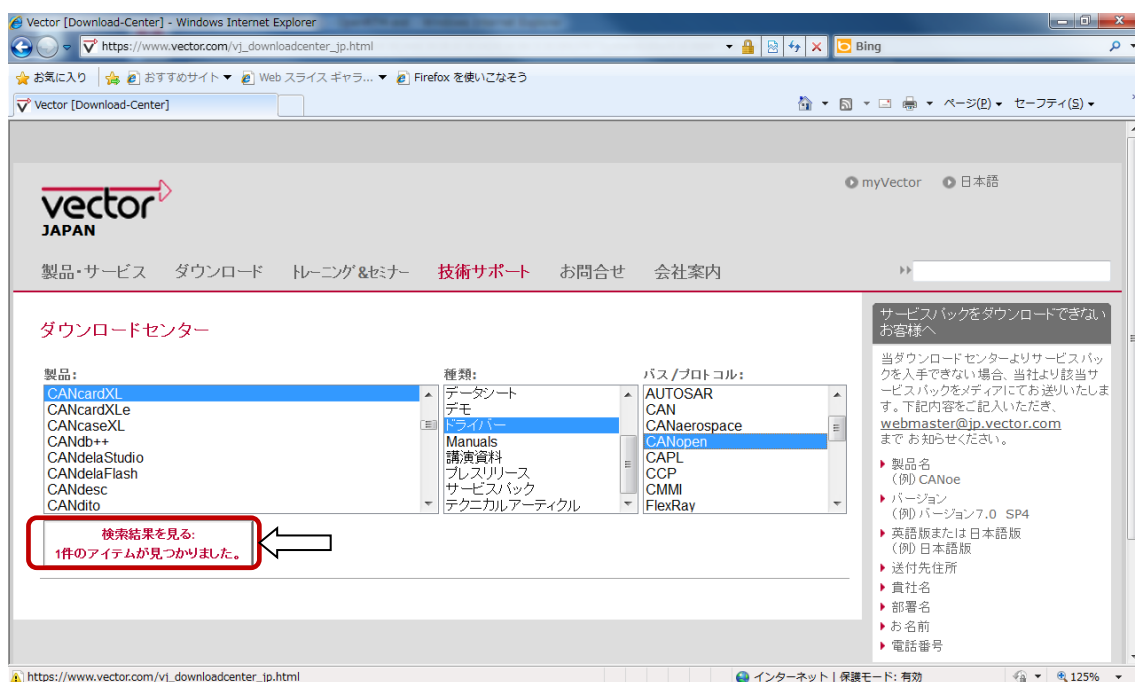


図 3 CANcardXL ダウンロード画面

すると以下のような画面になるのでダウンロード(日本国内用)をクリックする。

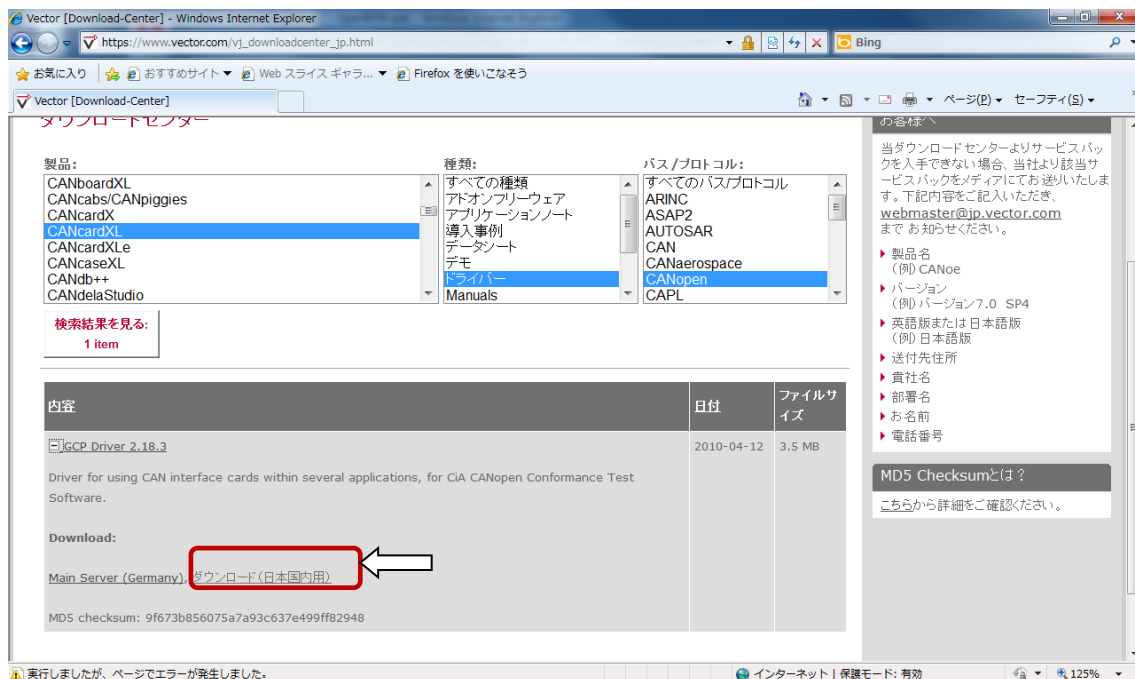


図 4 CANcardXL ダウンロード選択画面

ダウンロードし、実行すると以下の画面になるので Next をクリックする。

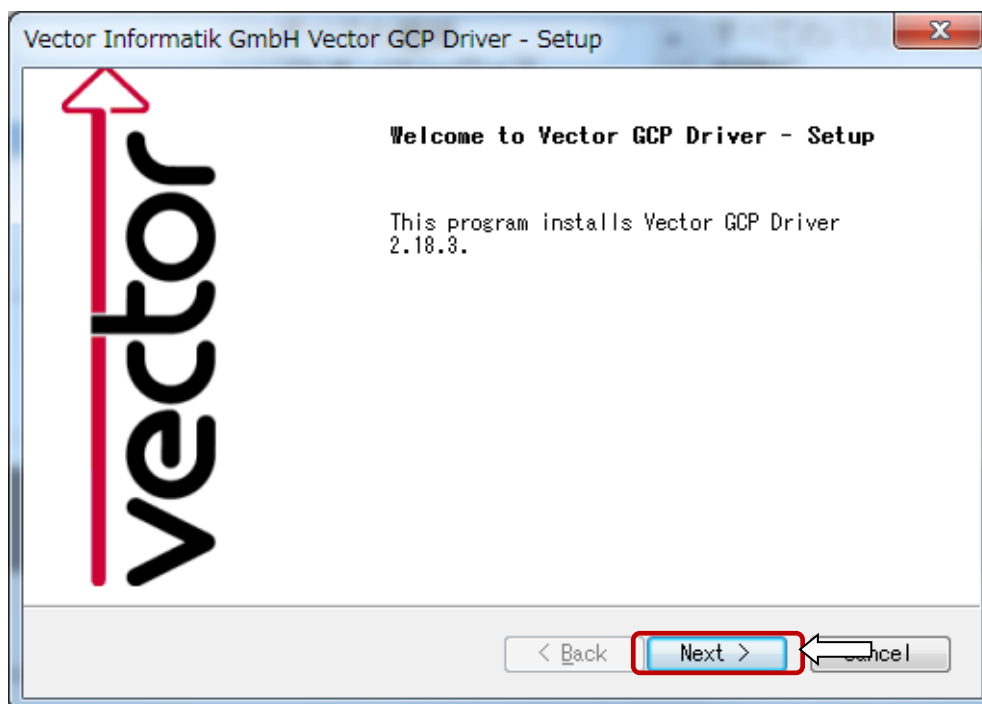


図 5 VectorDriver セットアップ画面

Vector Tool を選択し、Next をクリックする。

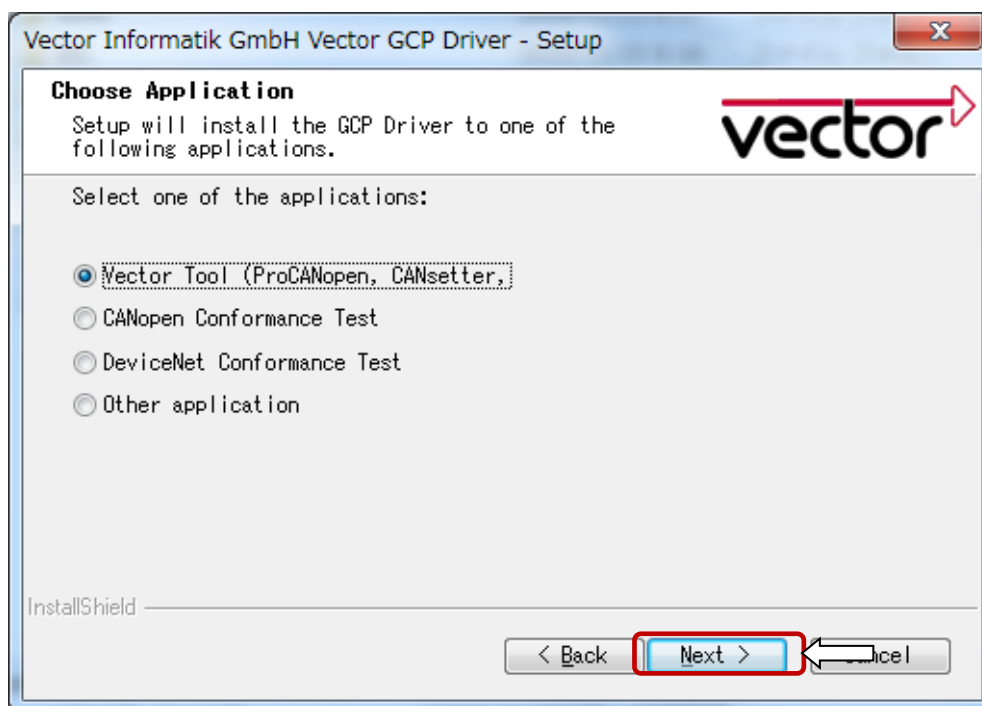


図 6 Vector Driver セットアップ選択画面

次に ProCANopen を選択し、Next をクリックする。あとは全部 Next をクリックすれば完了となる。

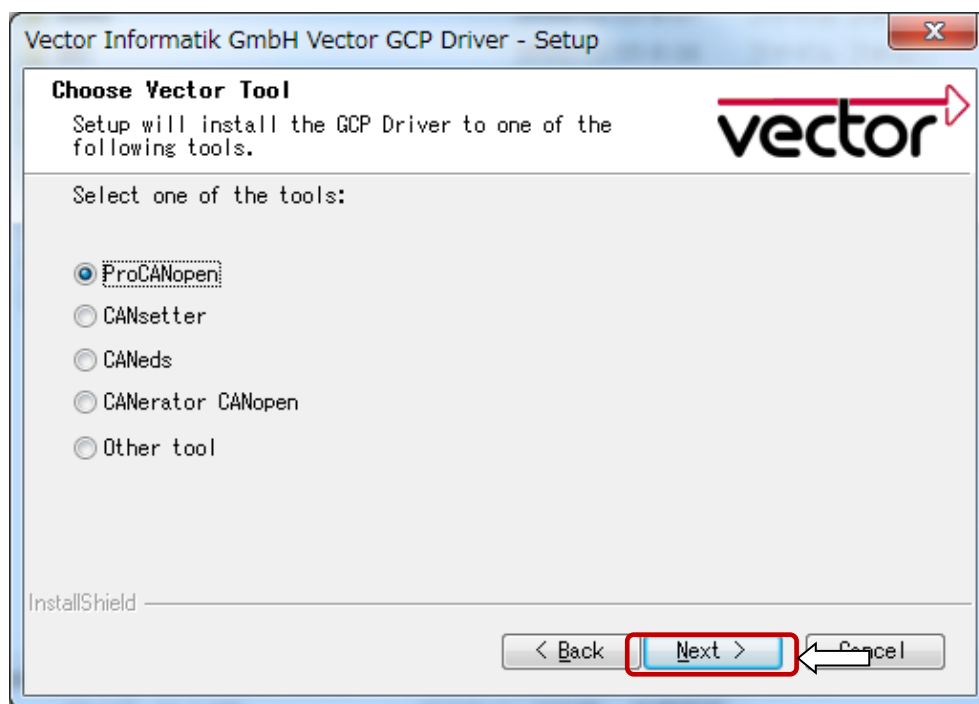


図 7 Vector Driver セットアップ完了画面

2.4. DCF の作成(Maxon Motor 社の EPOS の場合)

RTC-CANopen を利用する上で CANopen 製品の DCF が必要となる。DCF を作成する一例として Maxon Motor 社の EPOS を使用する場合について説明する。パラメータに関しては RTC-CANopen リファレンスマニュアル～RTC-CANopen の利用例～で記載してあるリファレンスロボットを例に設定をしていく。EPOS の DCF の作成には EPOS 用設定ソフトである EPOS Studio を用いる。

インストールしようとする以下画面になるので Next を押す。

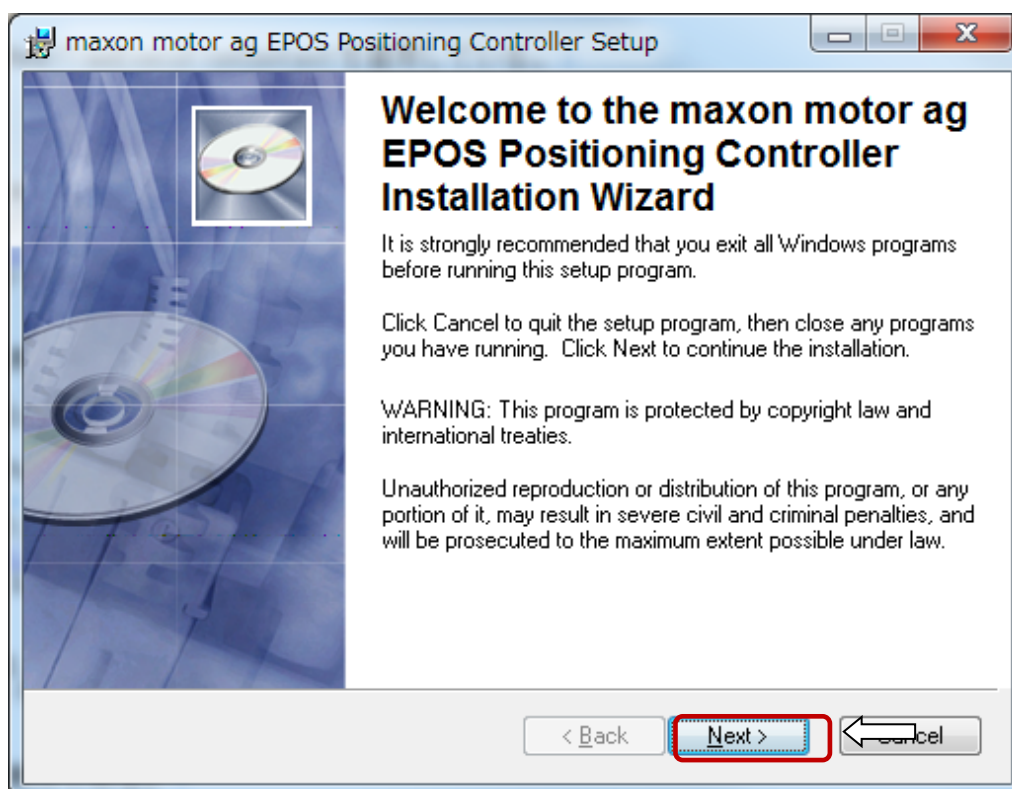


図 8 EPOS Studio セットアップスタート画面

順にセットアップを進めていくと最後に構成を編集するかと聞かれるが、標準の設定の **Typical** を押し、**Next** をクリックする。後はすべて **Next** で完了となる。

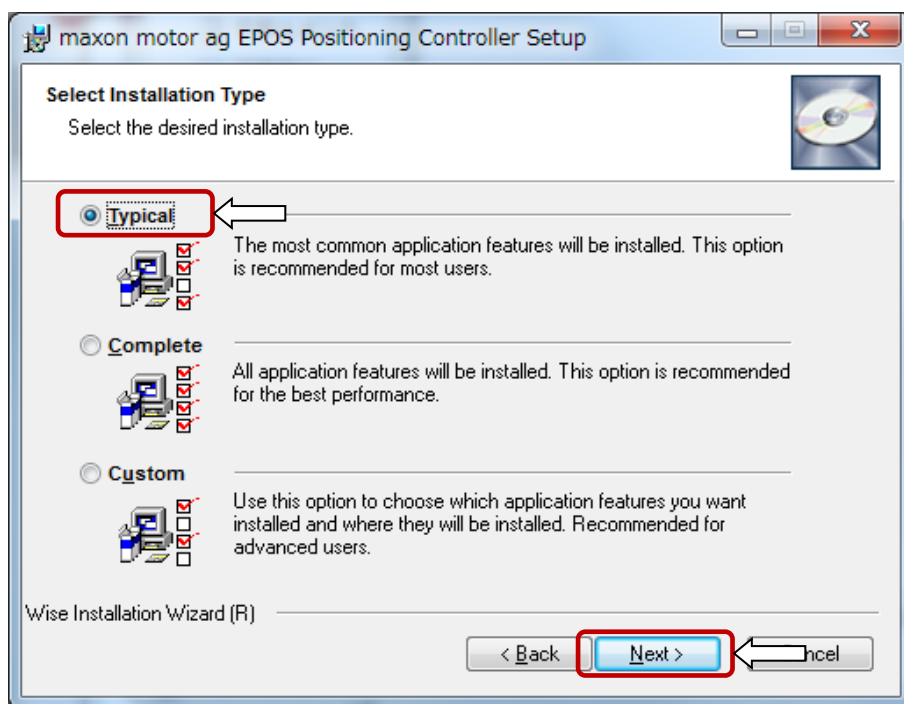


図 9 EPOS Studio セットアップ画面

EPOS Studio を立ち上げる．あらかじめ EPOS の電源を入れておく．以下のような画面が出るので EPOS2 Project を選択して「次へ」を押す．

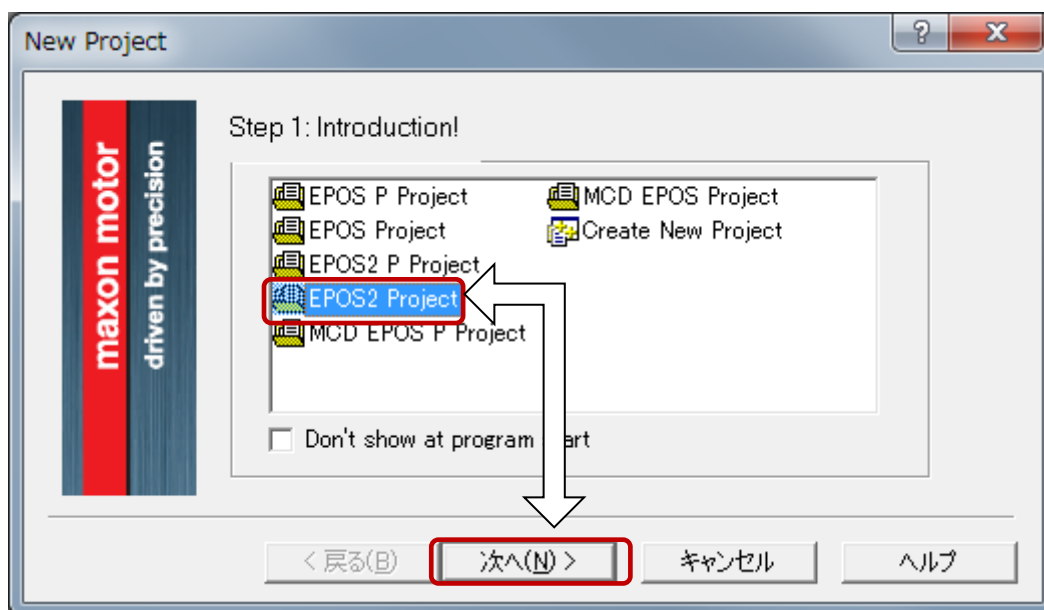


図 10 EPOS Studio 設定画面

次に保存するファイルを変更できるが，しない場合はそのまま「完了」となる．



図 11 EPOS Studio 設定画面 2

そして，USB で EPOS と接続する．接続がうまくいかない場合は何度か試す．

下にエラーが出る時があるが，Error の場所を右クリックし，Clear All Entries を選択する。

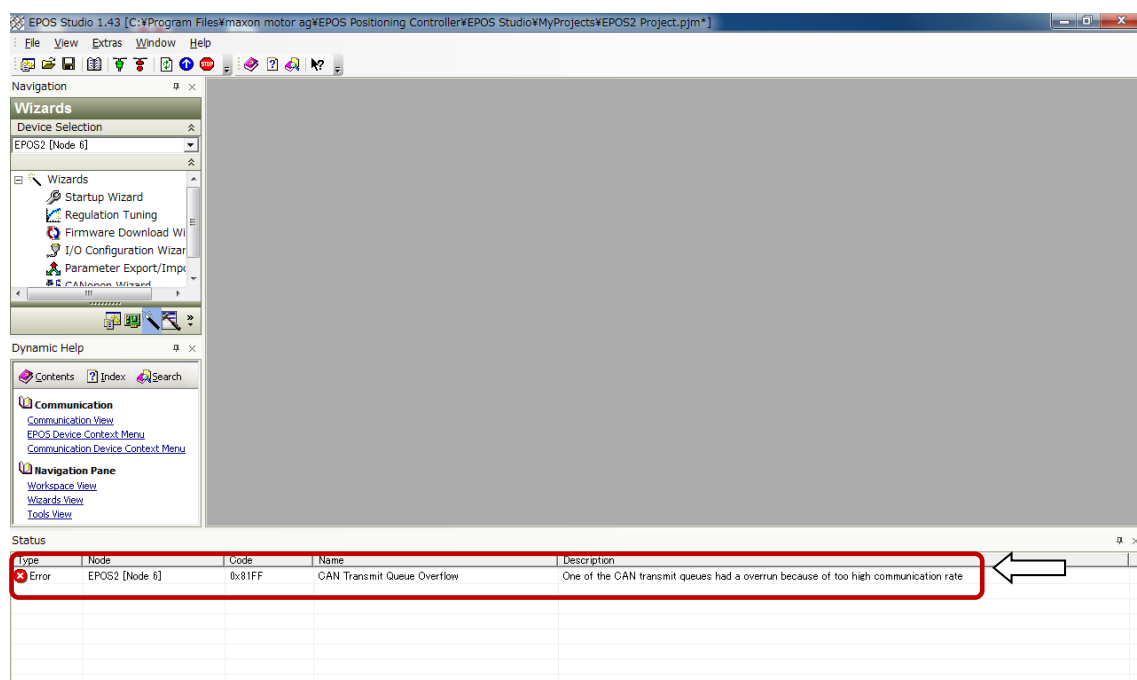


図 12 EPOS Studio Error 画面

Navigation ウィンドウの Wizard(ペンのようなマーク)をクリックし，Startup Wizard をクリックする。

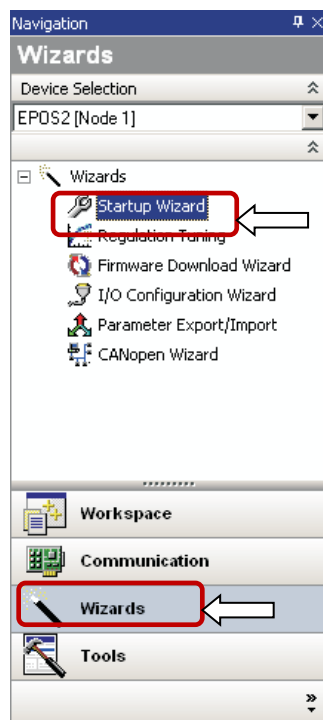


図 13 Starting Wizard

すると以下のような画面になるので、「Confirm that you've read the 'Getting Started' document」をクリックし、「次へ」を押す。

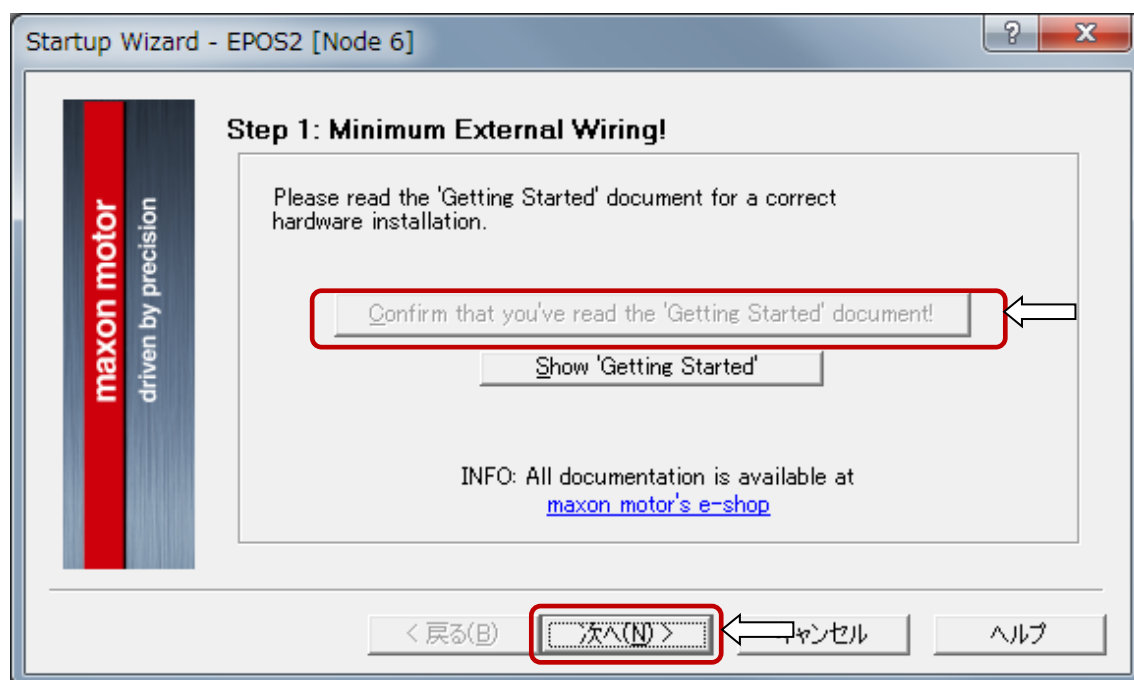


図 14 EPOS Studio DCF 設定画面

USB 接続が正確に行われているかを確認し、ボタン”Search Communication Setting”をクリックすると、使用されている USB ポートとボーレートが検出される。

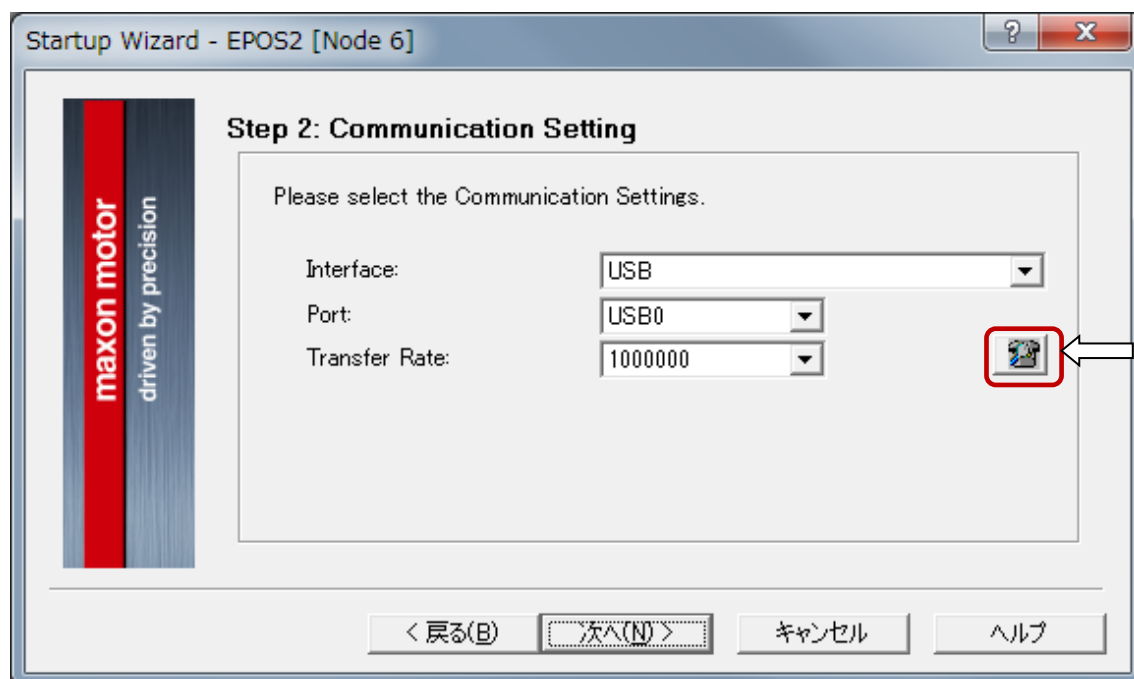


図 15 EPOS Studio DCF 設定画面 2

通信設定が検出された結果は以下のようになるのでそのまま **OK** を押し、「次へ」を選択する。

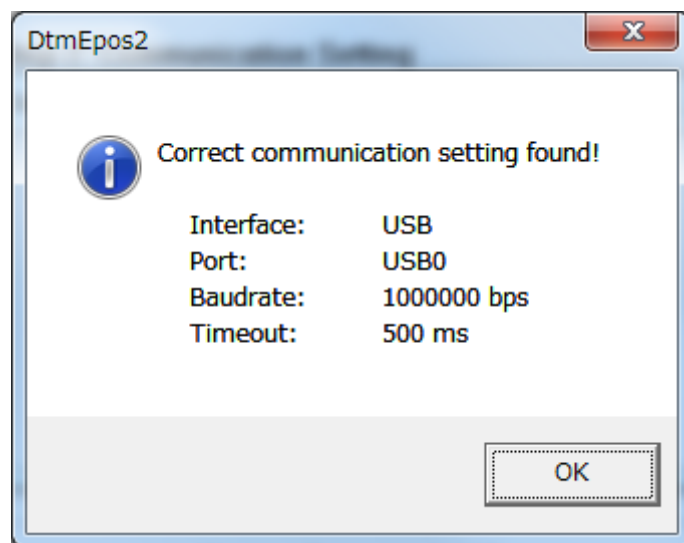


図 16 EPOS Studio 設定画面 3

使用するモータのタイプを選択する。ここでは DC モータを選択する。

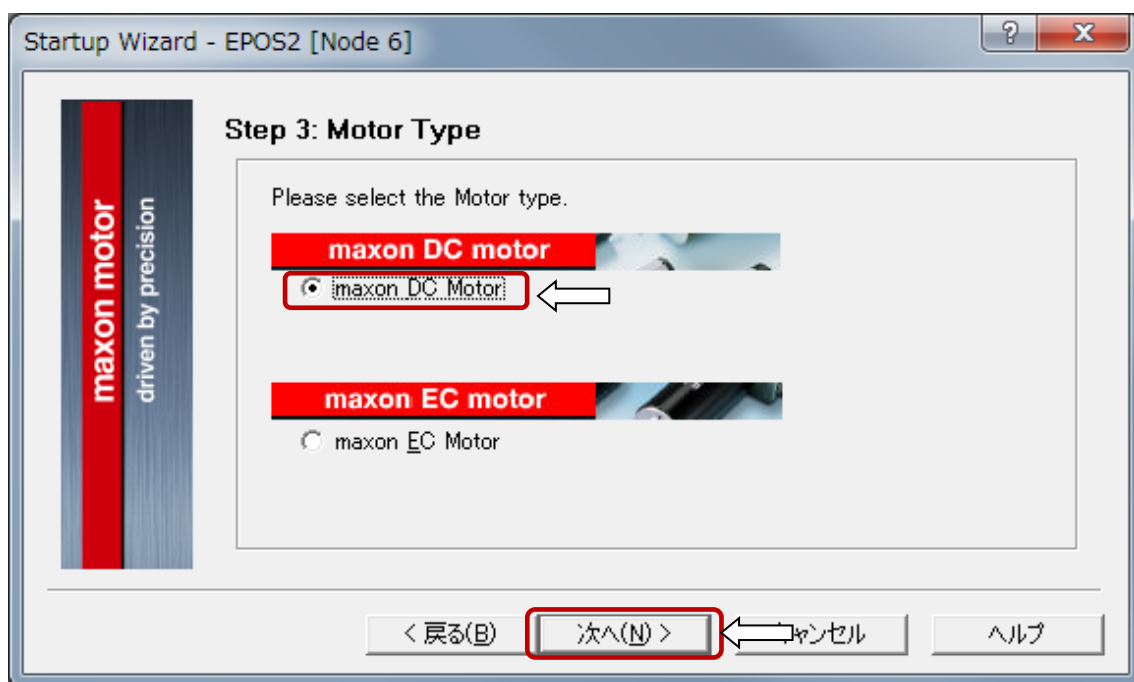


図 17 EPOS Studio DCF 設定画面

次にエンコーダのタイプを選択する。ここでは 3ch を選択する。

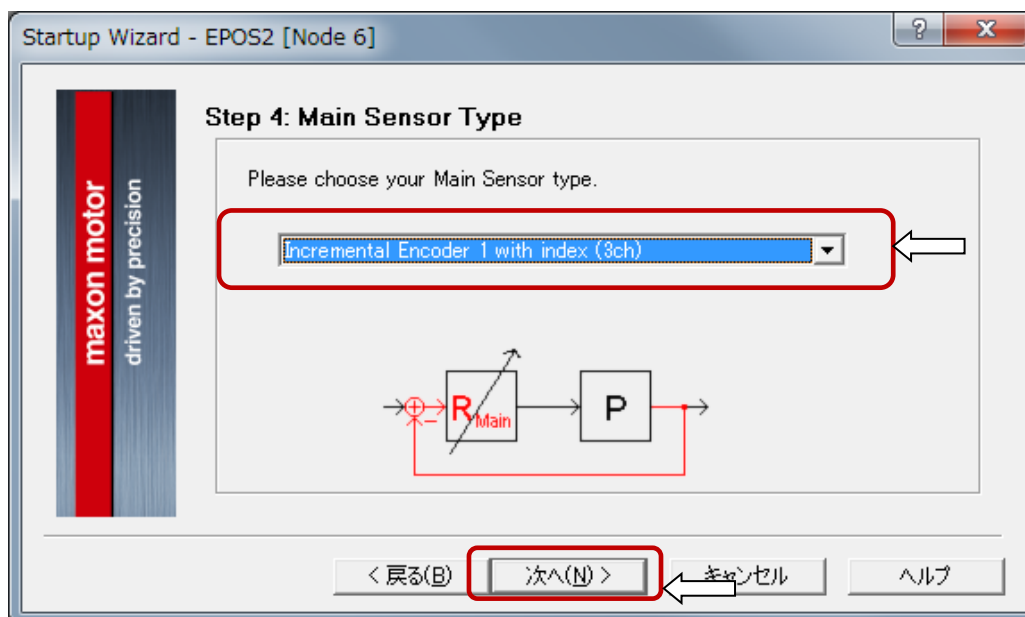


図 18 EPOS Studio 設定画面 5

モータのデータを入力する。今回のモータのデータは

最大許容範囲(Max. Permissible speed):9800rpm

公称電流(Nominal Current):853mA

最大連続電流(Max. Continuous current):1706mA

巻線熱定数(Thermal time constant winding):8.3s であるので以下のように入力して
「次へ」をクリックする。

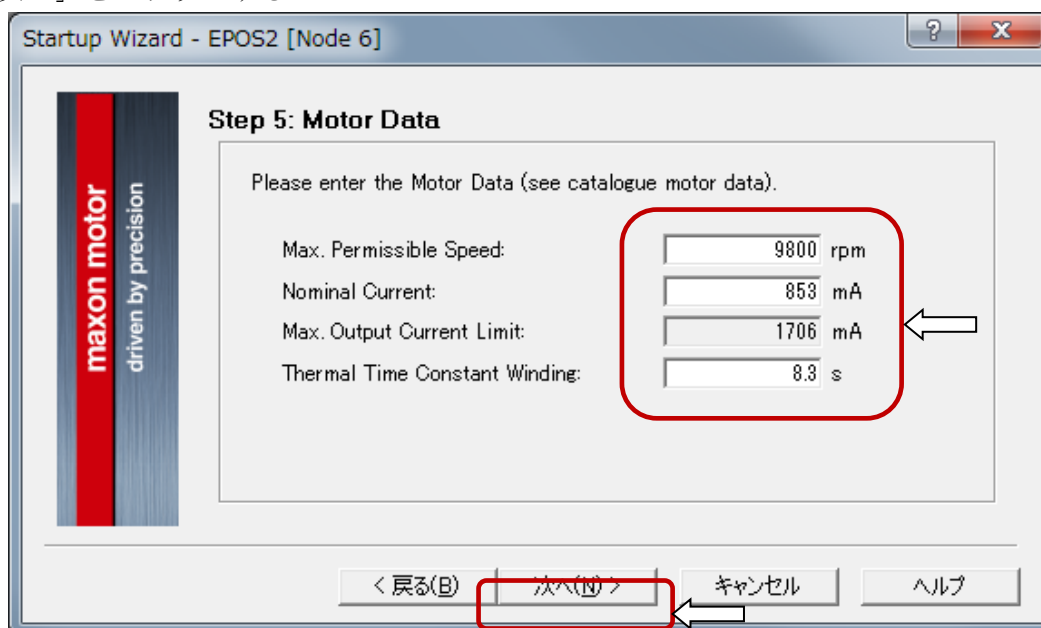


図 19 EPOS Studio 設定画面 6

エンコーダの分解能を入力する。エンコーダの分解能は 512 と入力し、「次へ」をクリックする。

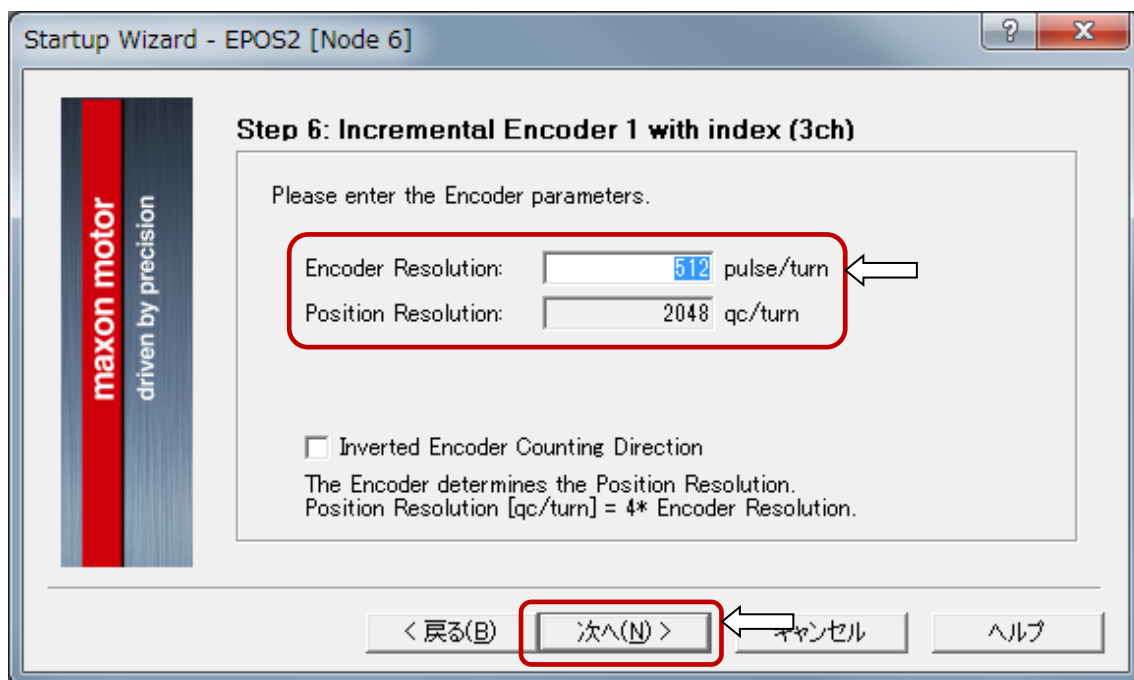


図 20 EPOS Studio DCF 設定画面 7

入力位置制御のときの位置ずれの許容範囲を設定する。この許容範囲を超えるとエラーが発生する。ここはデフォルトの 2000 に設定する。

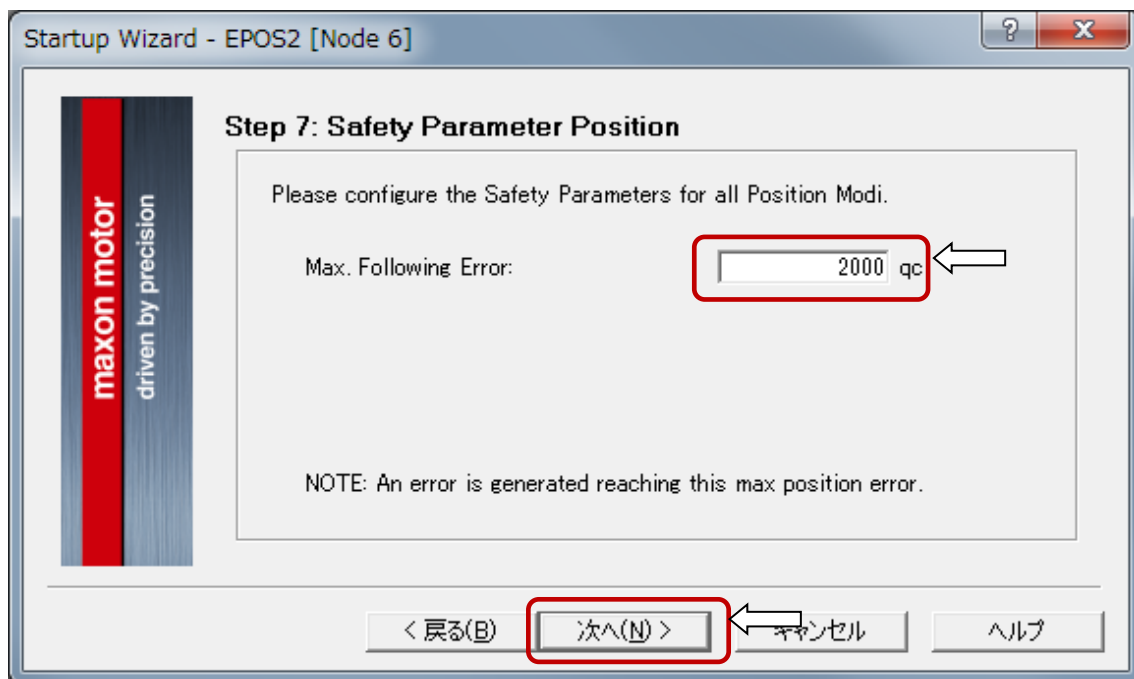


図 21 EPOS Studio DCF 設定画面 8

設定した値の確認画面になる。設定に間違いがあれば「戻る」で修正することができる。
設定が正しければ、完了を押す。

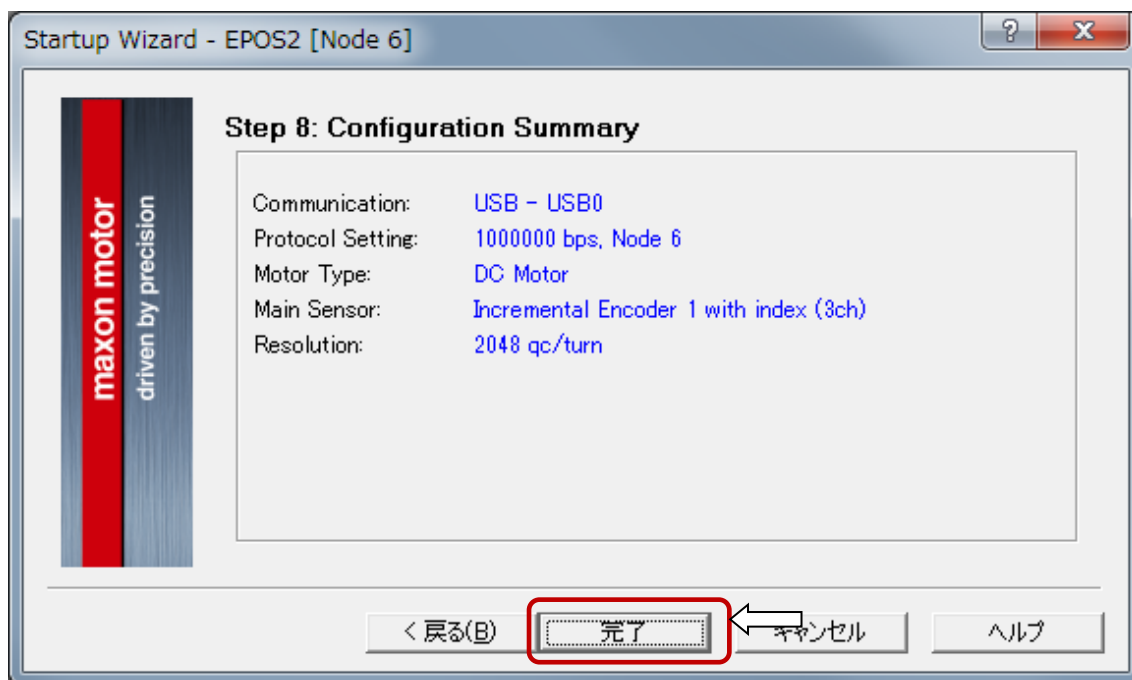


図 22 EPOS Studio 設定完了画面

「はい」を押せば、パラメータは EPOS に保存される。

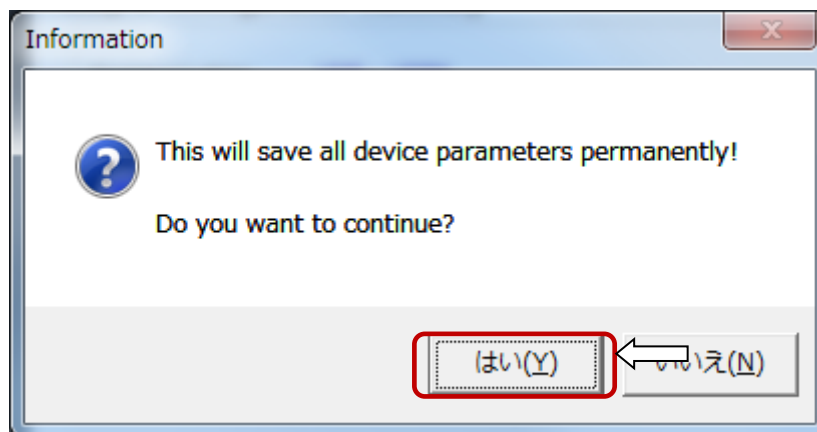


図 23 EPOS Studio 設定完了画面 2

EPOS2 24/5 は制御ゲインのオート・チューニング機能を有している。よって、電流、速度、位置の制御ゲインを自動的に調整することができる。(オート・チューニングの値が最適値とは限らない。以下に手順を説明する。

まず、Navigation ウィンドウの Wizards をクリックし、Regulation Turning を選択する。

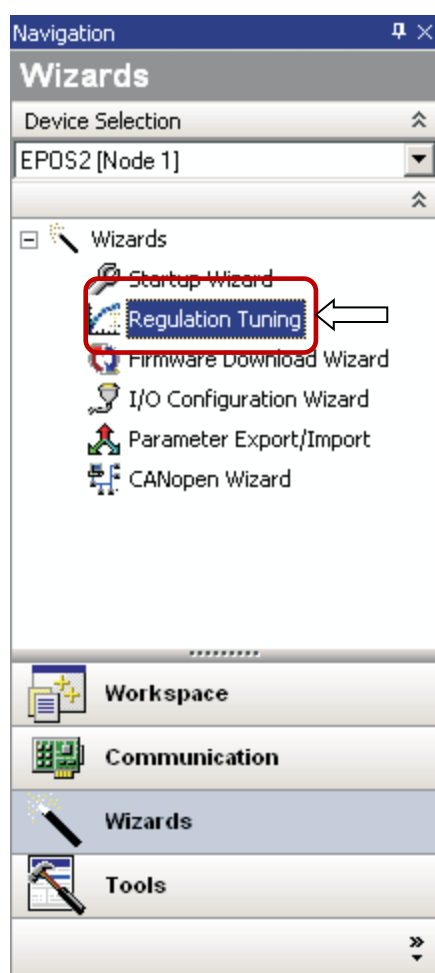


図 24 EPOS Studio Regulation Tuning

オート・チューニングを行う場合は **Auto Tuning** を、個別に制御値を設定する場合は **Expert Tuning** を選択する。

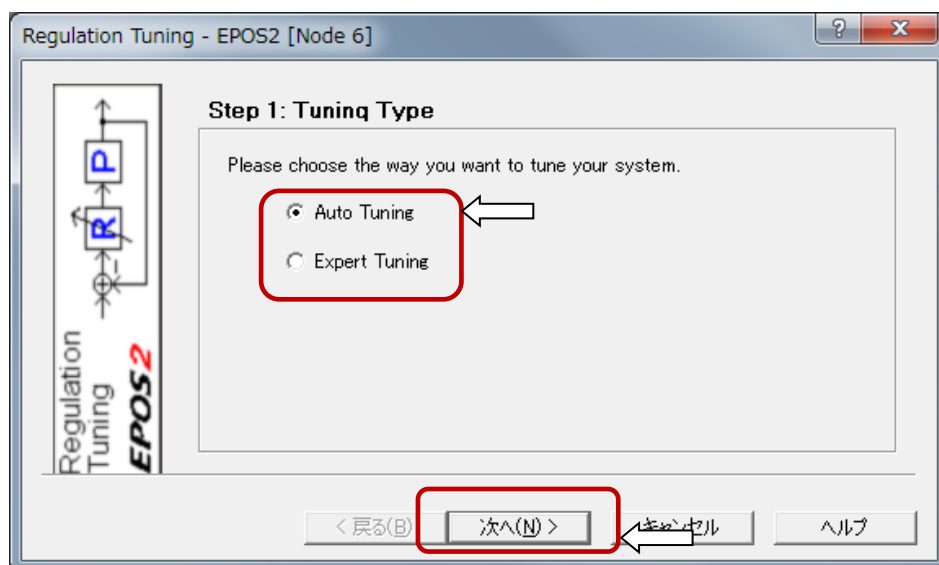


図 25 Regulation Turning 選択画面

Auto Regulation Tuning のウィンドウが表示される。 **Start** を押してオート・チューニングを開始する。

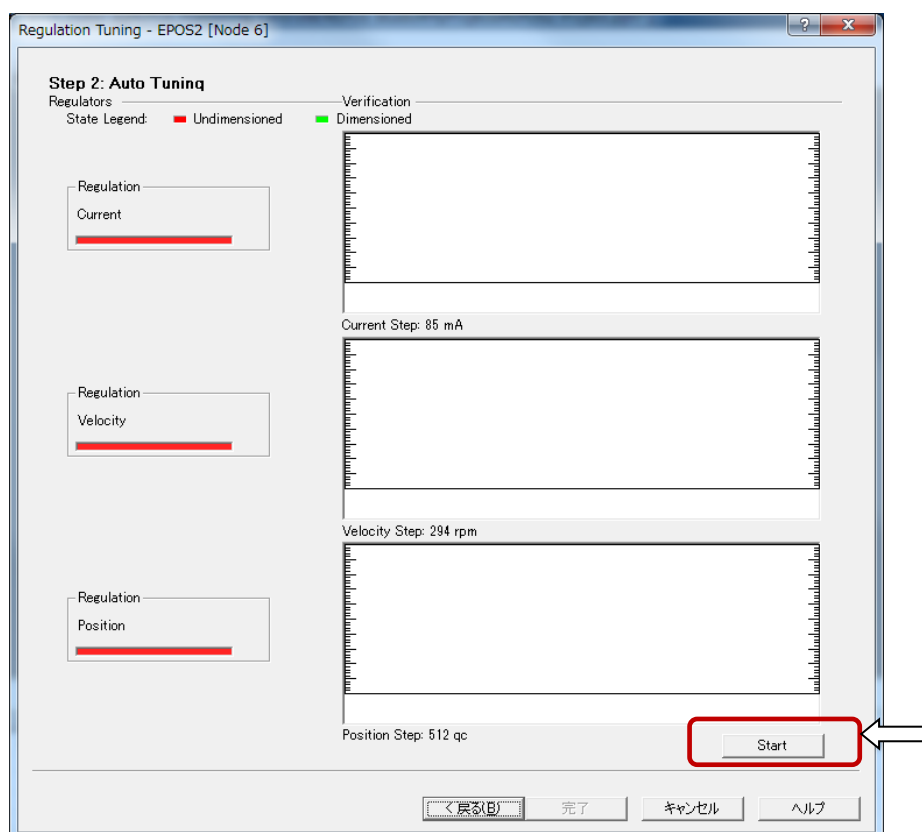


図 26 Regulation Tuning ウィンドウ画面

モータの軸がフリーであるかを聞かれるので、モータの軸をフリーにする。

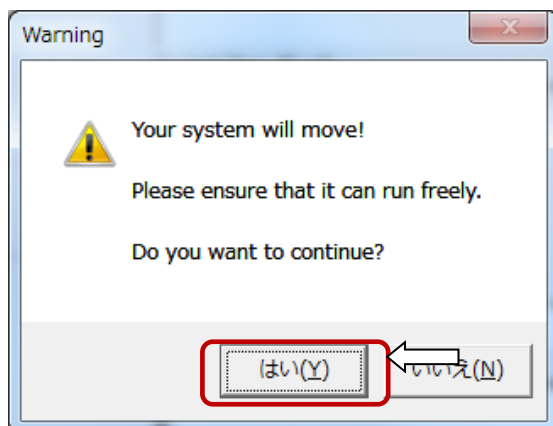


図 27 Regulation Tuning ウィンドウ画面

もしエラーが発生した場合は OK を押して、Expert Tuning を選択する。

次に TPDO 及び RPDO のマッピングを行う。マッピングとは 8 バイトある PDO の送信すべき情報と受信した格納場所を設定することである。

まず初めに Wizard>CANopen Wizard をクリックする。

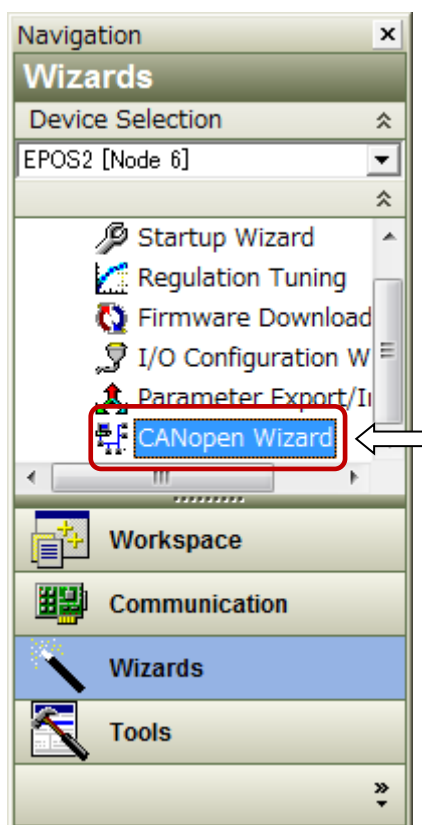


図 28 CANopen Wizard

すると以下のような画面になるので「次へ」を選択する。

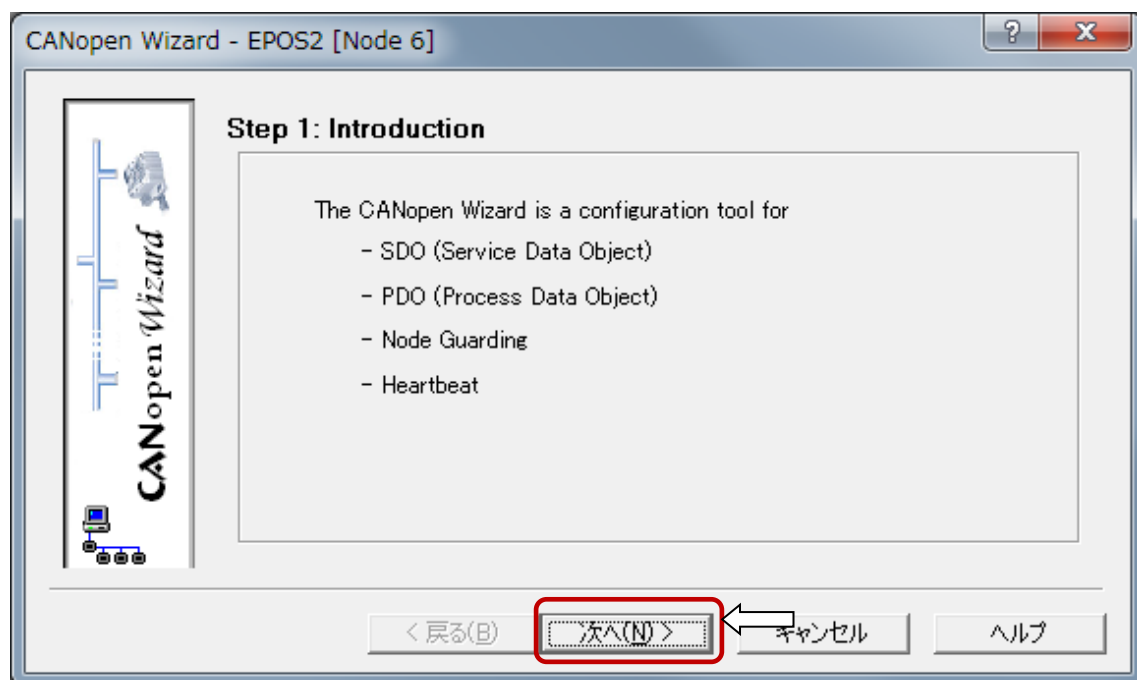


図 29 マッピング設定画面

COB-ID とは CAN メッセージの ID のことで、ネットワーク上のメッセージを識別する際に用いられる。しかしここではデフォルトのままで OK であるので「次へ」を選択する。

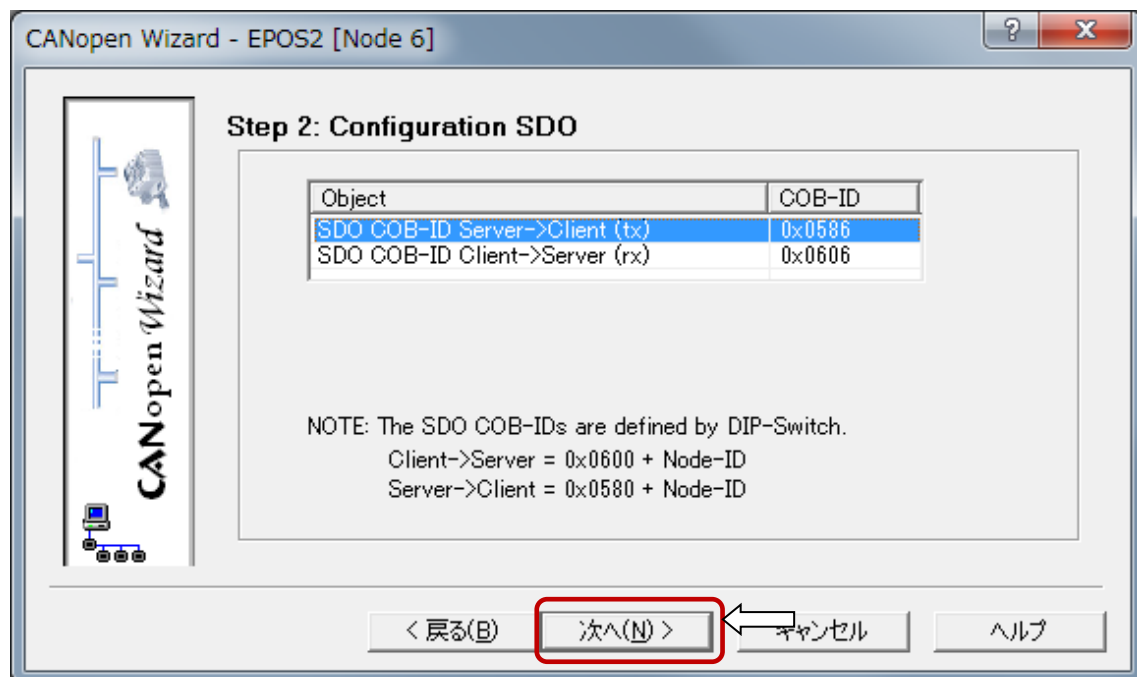


図 30 Configuration SDO

PDO のコンフィギュレーションの画面になる。確認したら、「次へ」をクリックする。

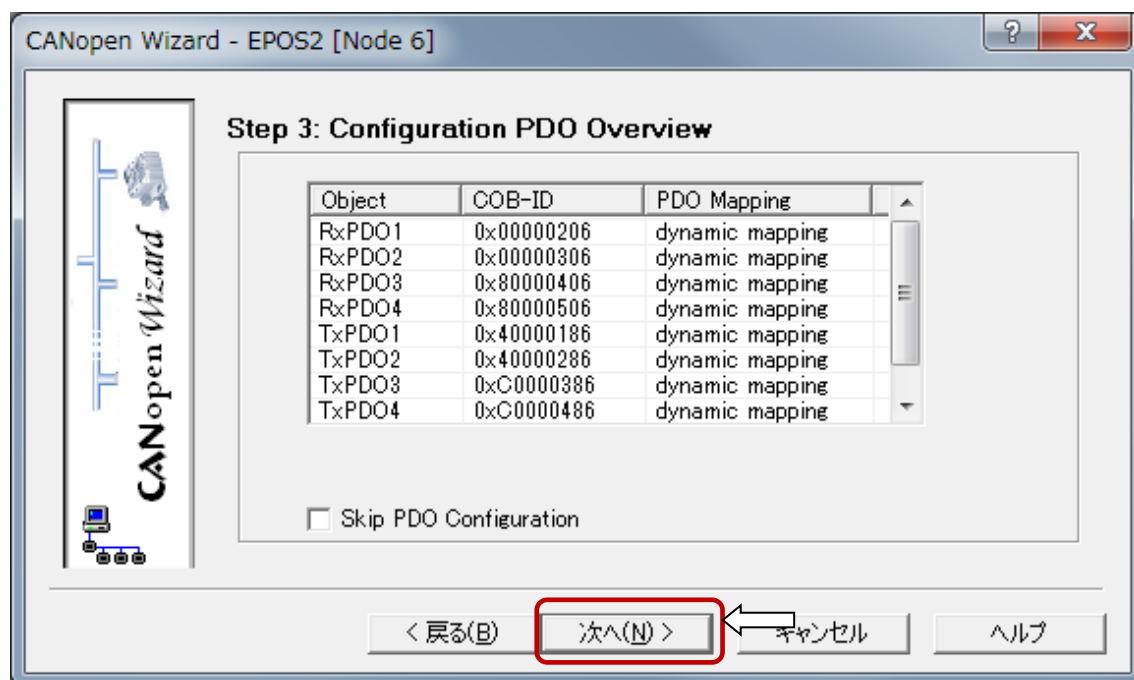


図 31 Configuration PDO Overview

デフォルトの COB-ID を保存したい場合は Restore Default COB-IDs をクリックする。
「次へ」をクリックする。

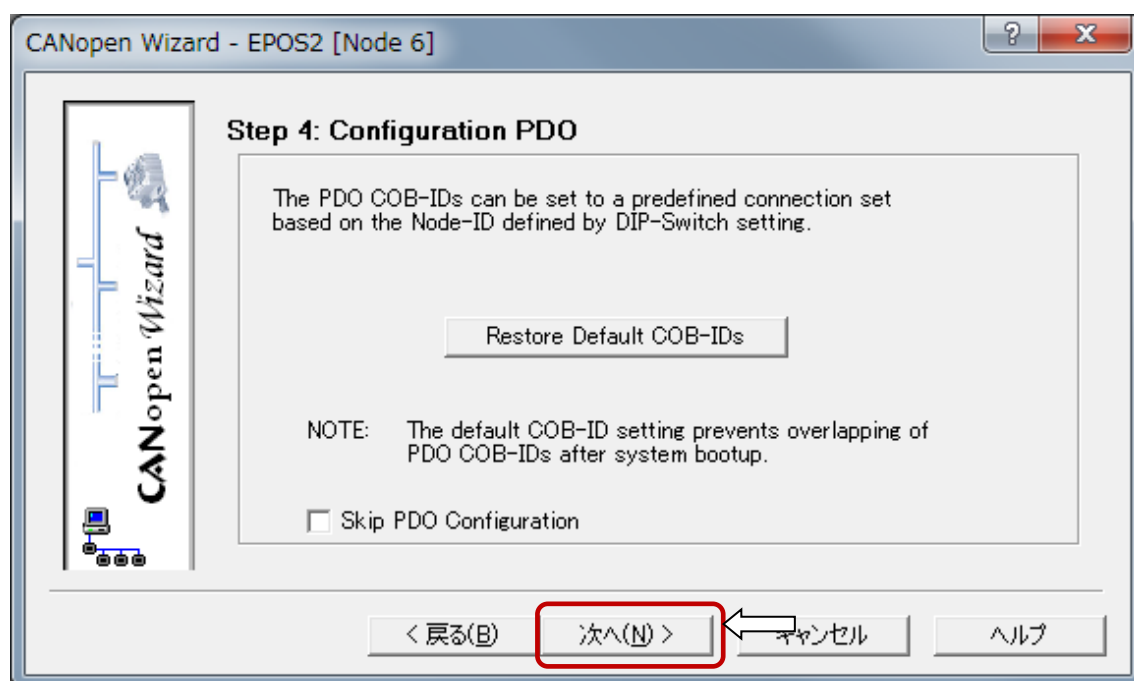


図 32 Configuration PDO

ここでは COB-ID をデフォルトのまま、Transmission Type(伝達方式)を Asynchronous(非同期)に設定する。

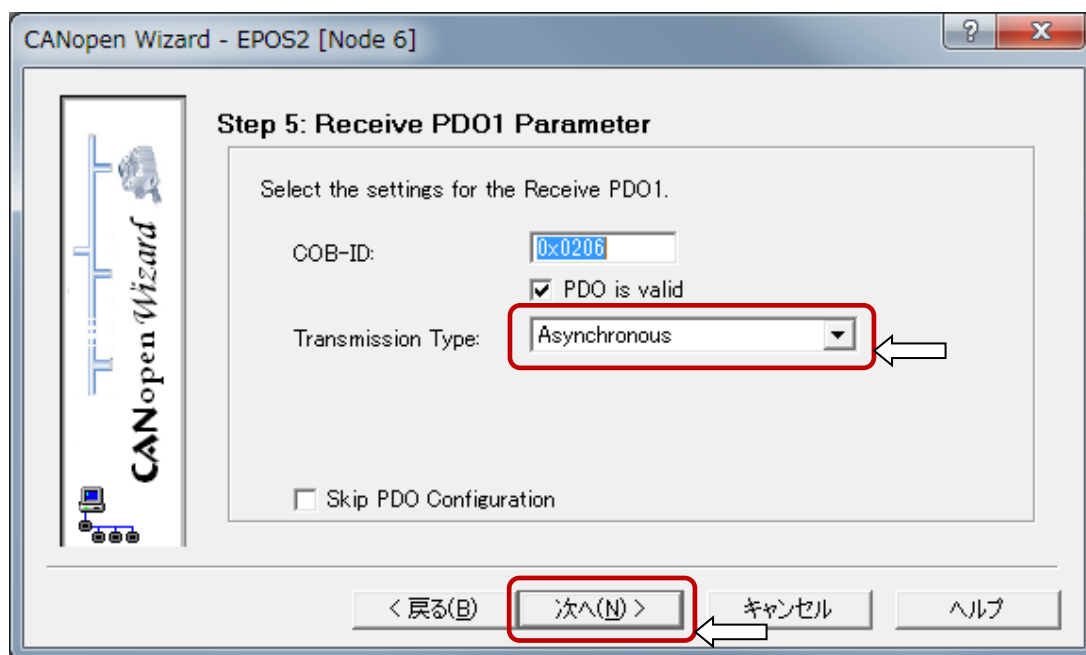


図 33 PDO Parameter

RPDO のマッピングを設定する。まず、Change Mapping をクリックし、ControlWord(2Byte)を選択する。選択し終わったら、「次へ」をクリックする。

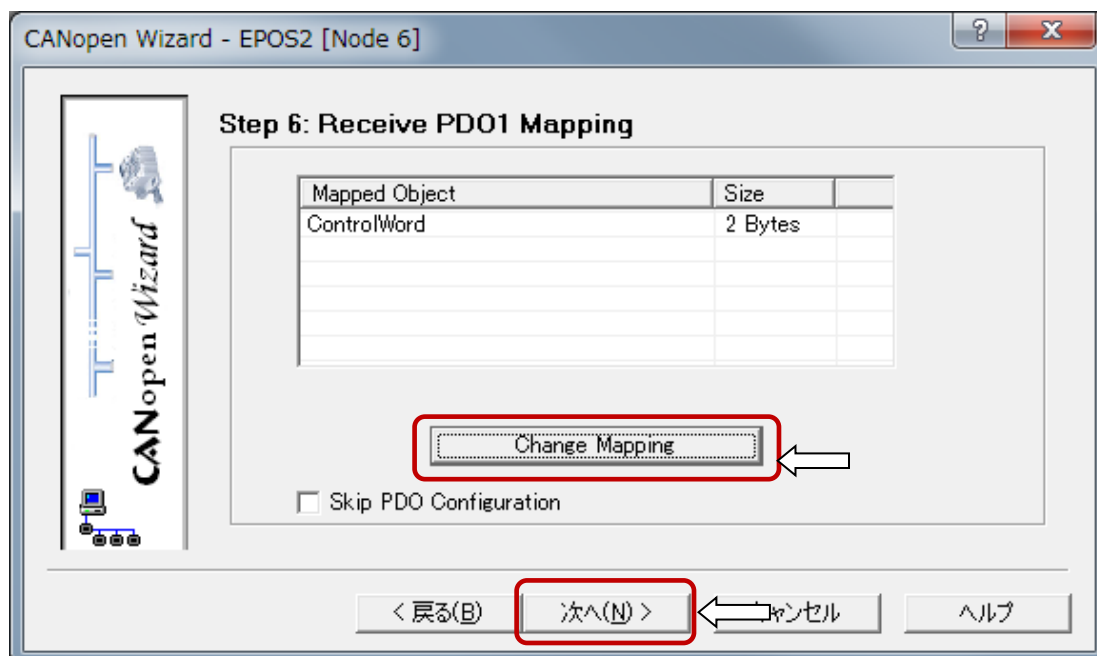


図 34 PDOmapping

次の画面も同様に Asynchronous(非同期)に設定する.

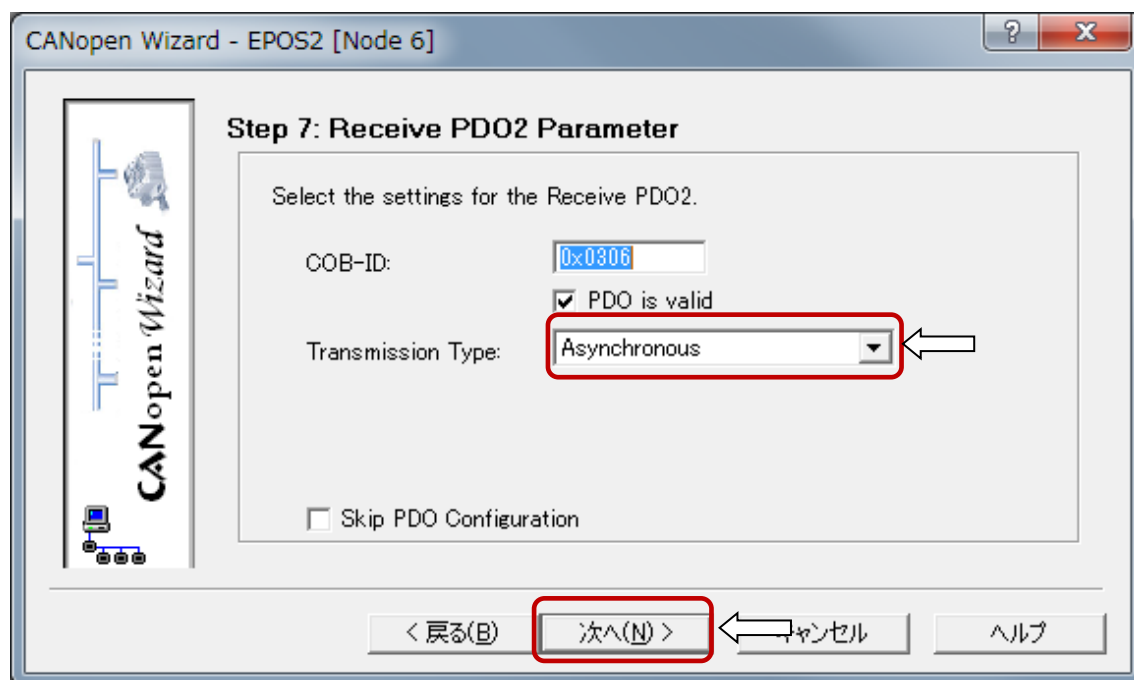


図 35 PDO Parameter2

次のマッピング(RPDO2)では ControlWord(2Bytes), Modes of Operation(1Byte), TargetVelocity(4Bytes)をマッピングする.

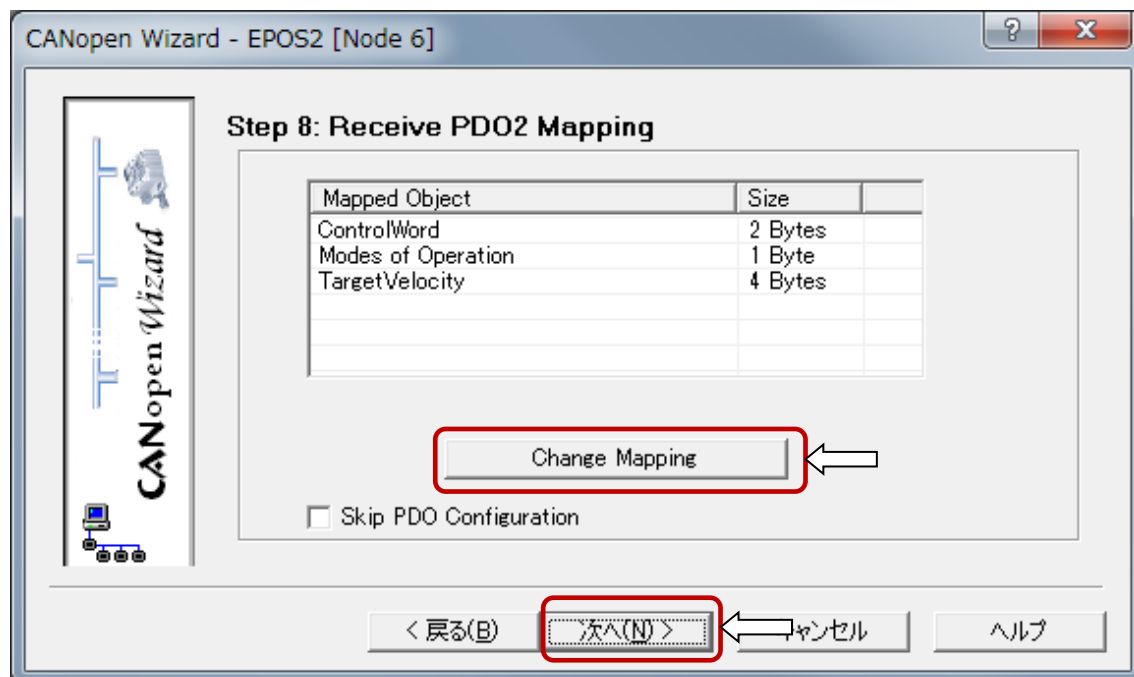


図 36 PDOmapping2

同様の手順で RPDO3 でも Asynchronous(非同期)に設定する.

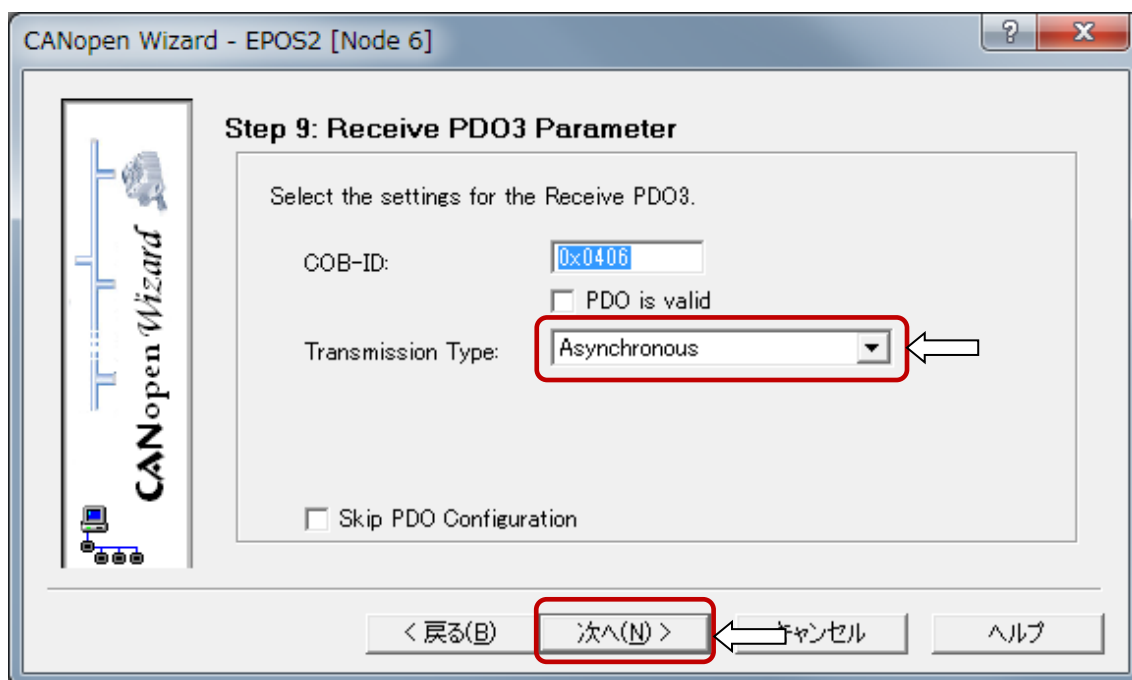


図 37 PDO Parameter3

RPDO3 では ControlWord(2Bytes), Modes of Operation(1Byte), TargetPosition(4Bytes)をマッピングする.

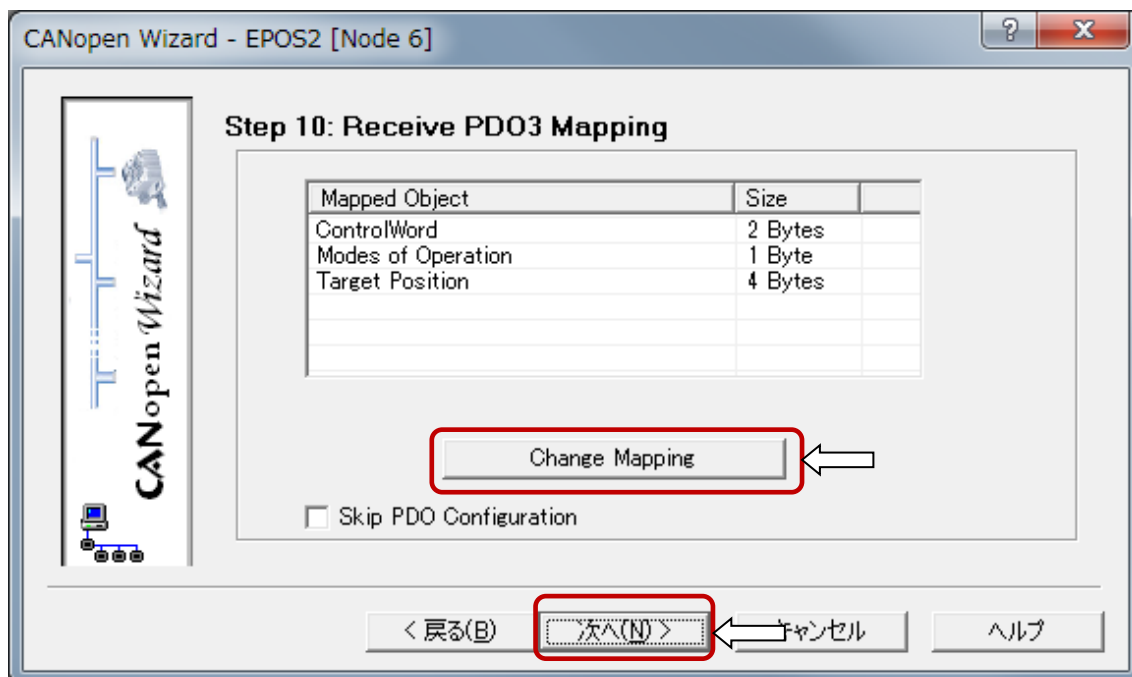


図 38 PDOmapping3

同様に RPDO4 を Asynchronous(非同期)に設定する.

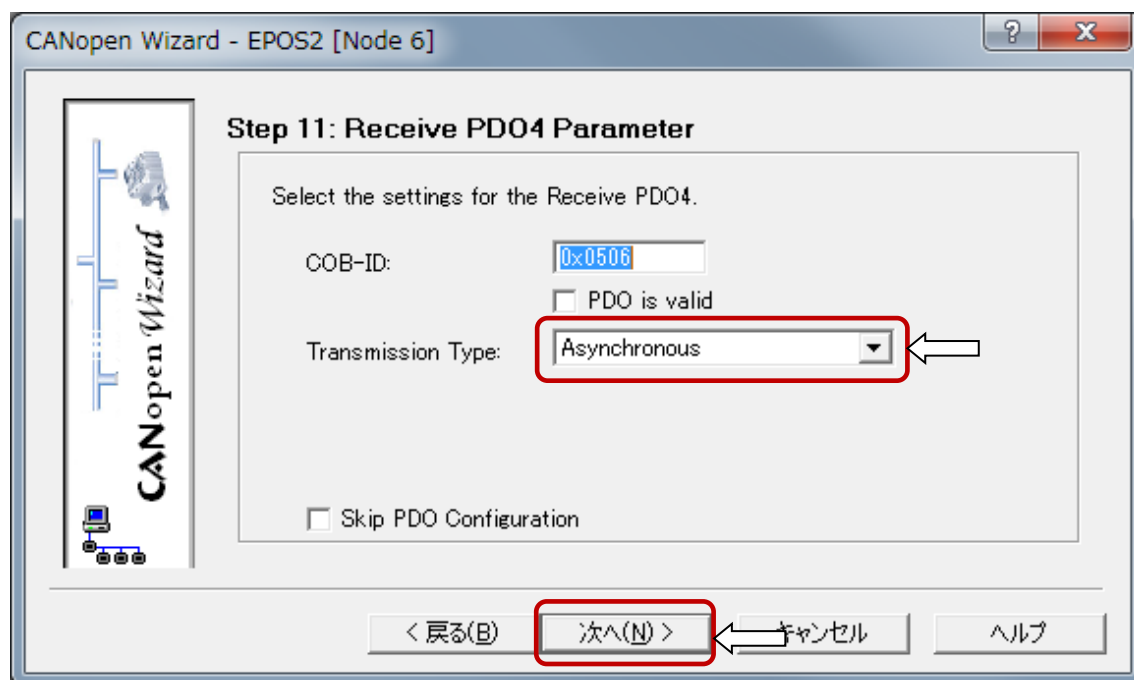


図 39 PDO Parameter4

RPDO4は ControlWord(2Bytes), Modes of Operation(1Byte), TargetVelocity(4Bytes)をマッピングする.

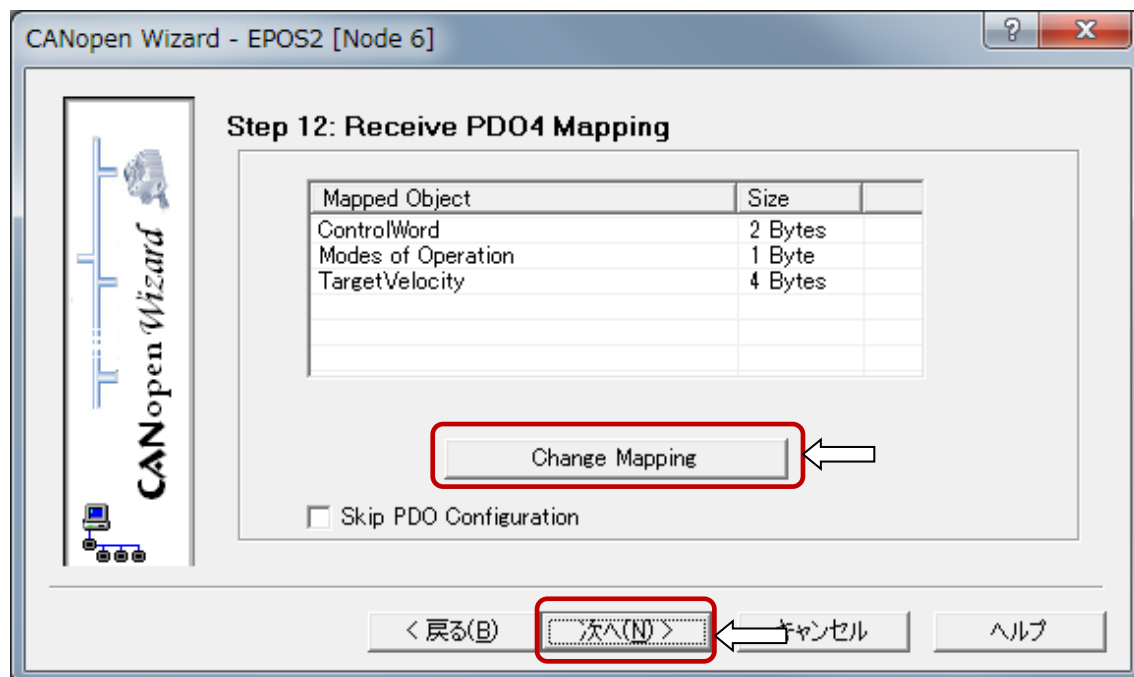


図 40 PDO Mapping4

次に TPDO のパラメータの設定を行う。伝達方式を Transmission Type(伝達方式)を Asynchronous(非同期)に設定する。

Inhibit Time(次の送信までの送信間隔)を 100.0ms に設定する。

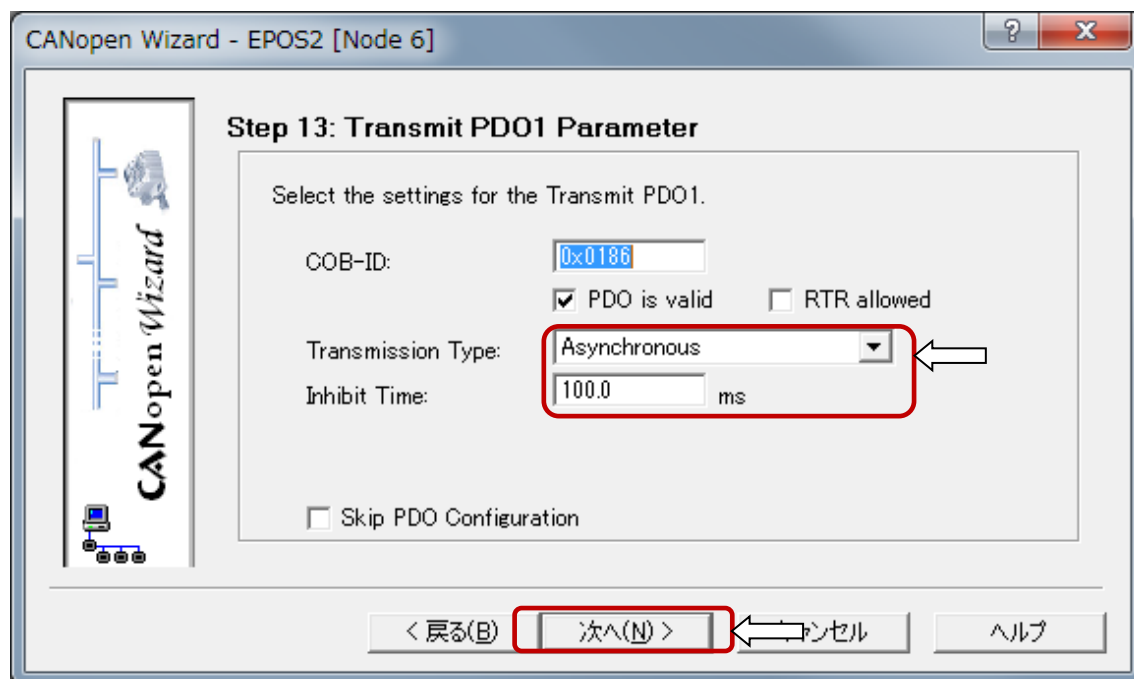


図 41 TPDO Parameter

TPDO1 のマッピングは StatusWord(2Bytes)を選択する。

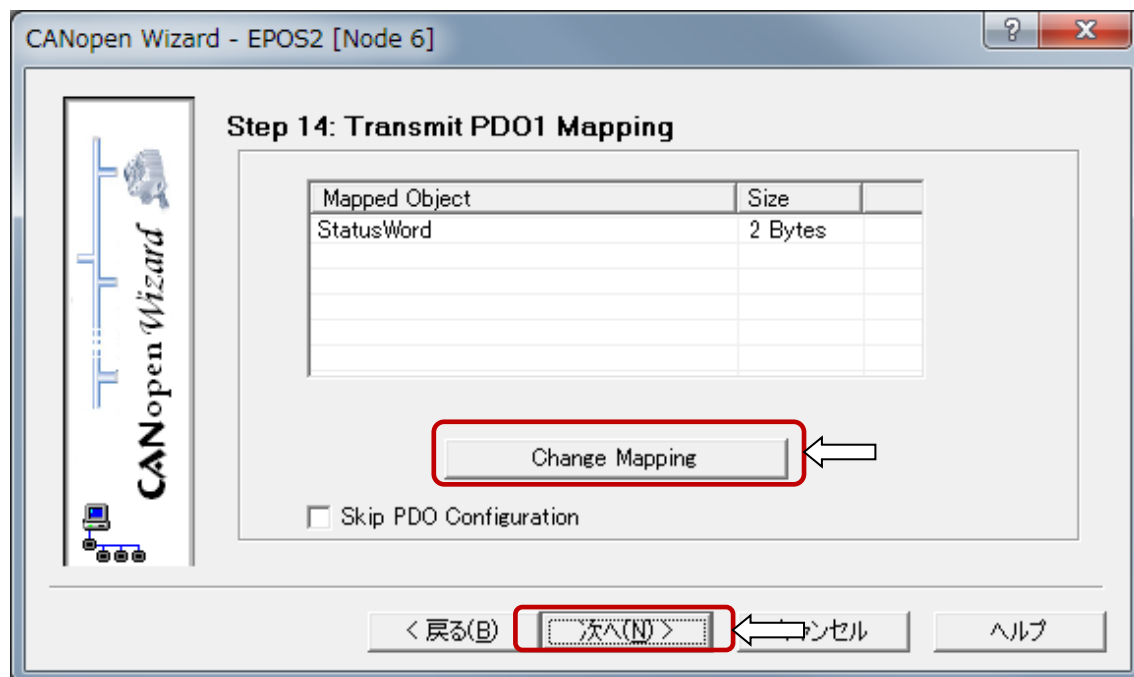


図 42 TPDO mapping

PDO2 も同様に Asynchronous(非同期), Inhibit Time(次の送信までの送信間隔)を 100.0ms に設定する.

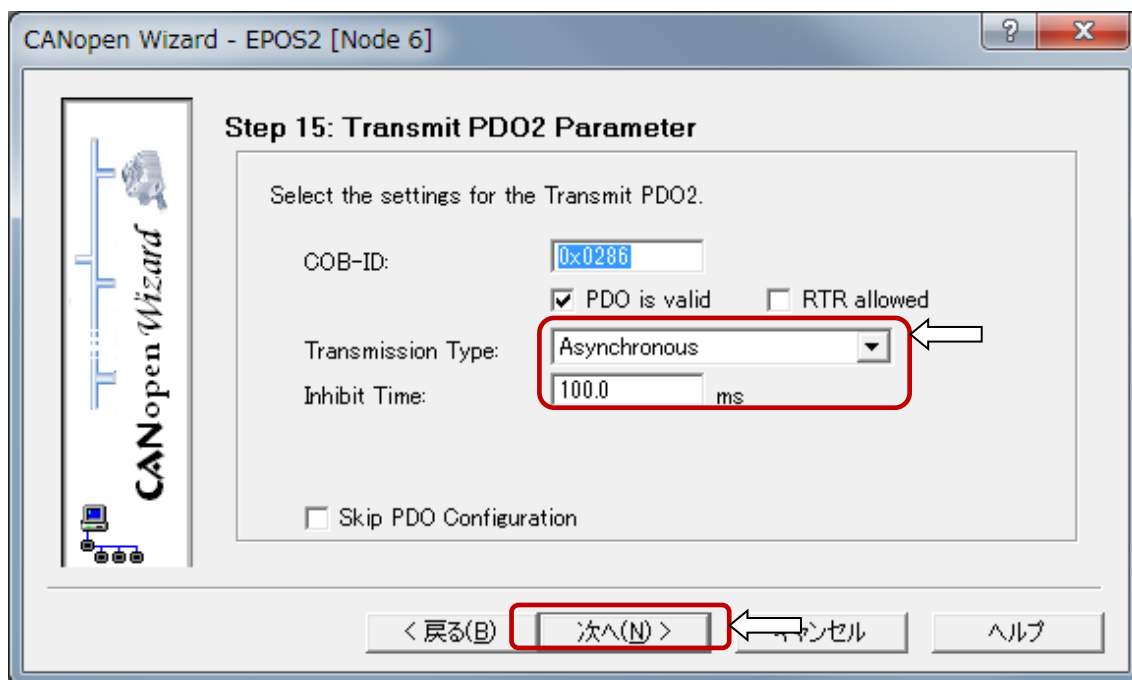


図 43 TPDO Parameter2

TPDO のマッピングは PositionActualValue(4bytes)を設定する.

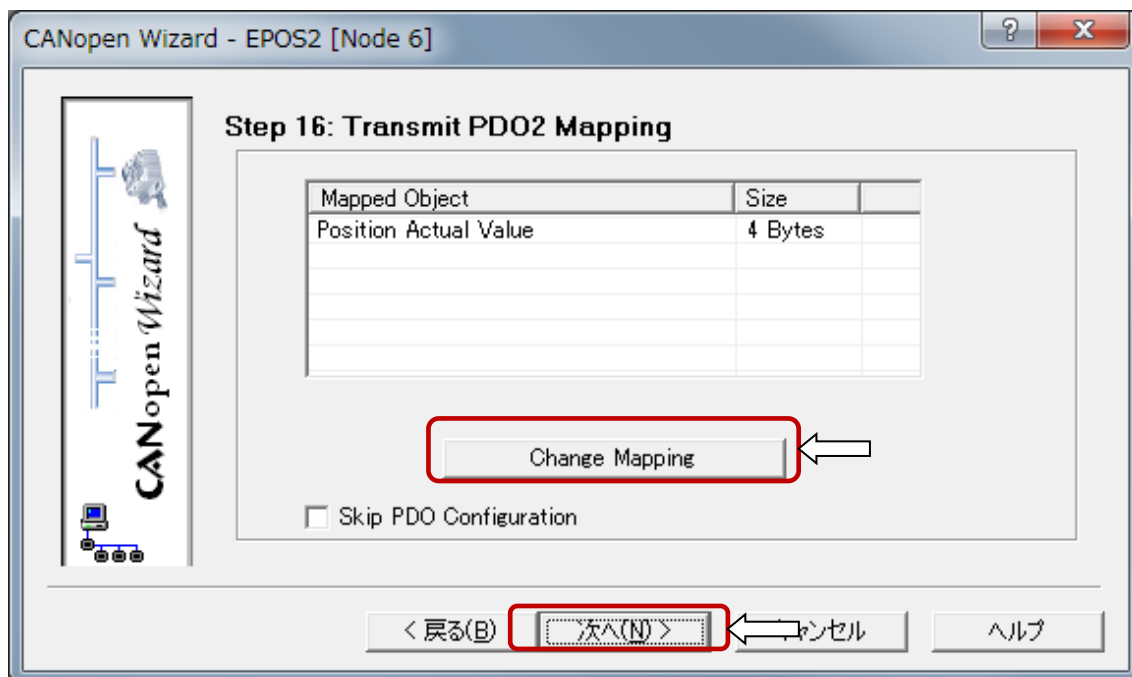


図 44 TPDO mapping2

PDO3 は Asynchronous(非同期), Inhibit Time(次の送信までの送信間隔)を 100.0ms に設定する.

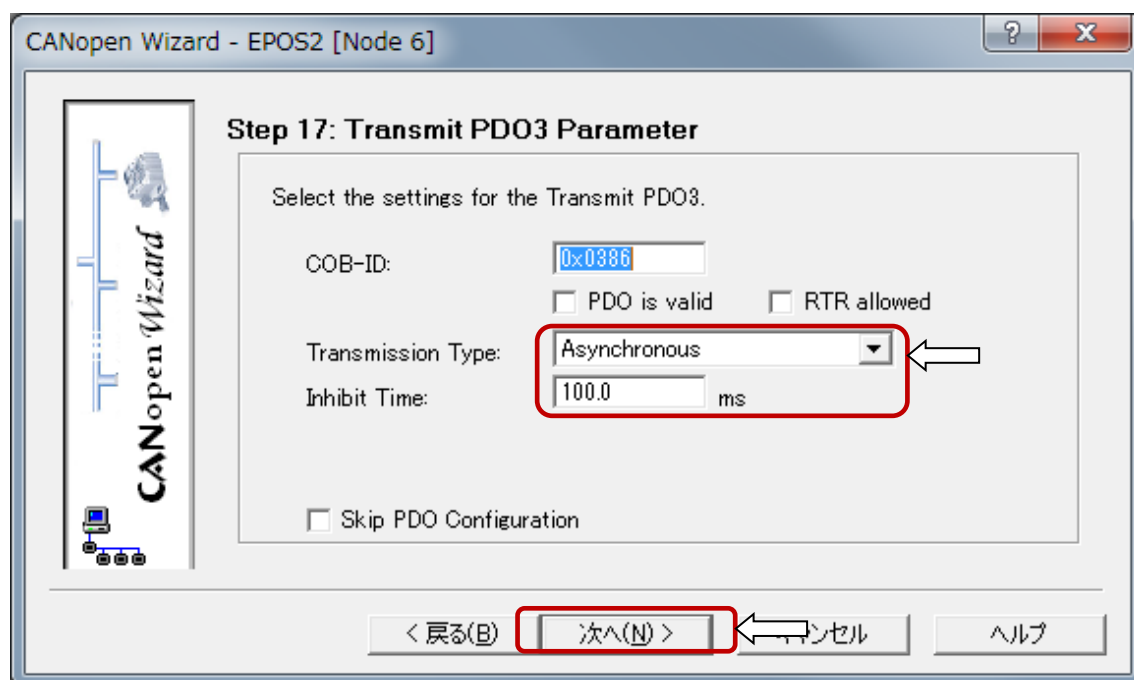


図 45 TPDO Parameter3

TPDO3 のマッピングは Position Actual Value(4Bytes), Velocity Actual Value(4Bytes)を設定する.

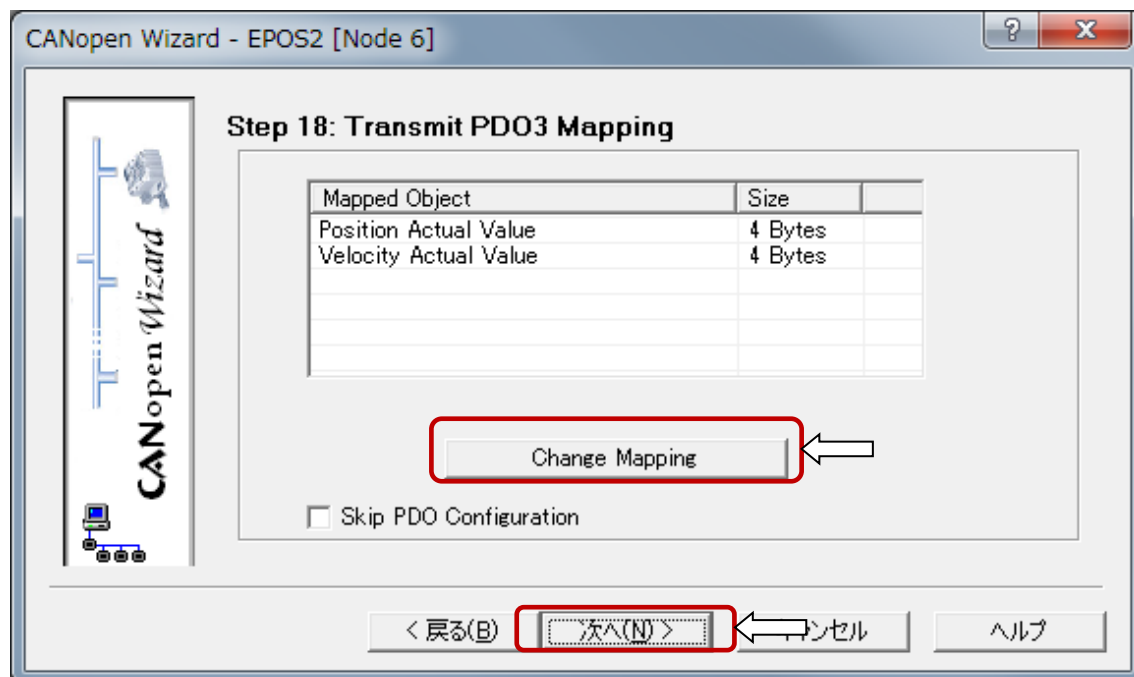


図 46 TPDO mapping3

PDO4 は Asynchronous(非同期), Inhibit Time(次の送信までの送信間隔)を 100.0ms に設定する.

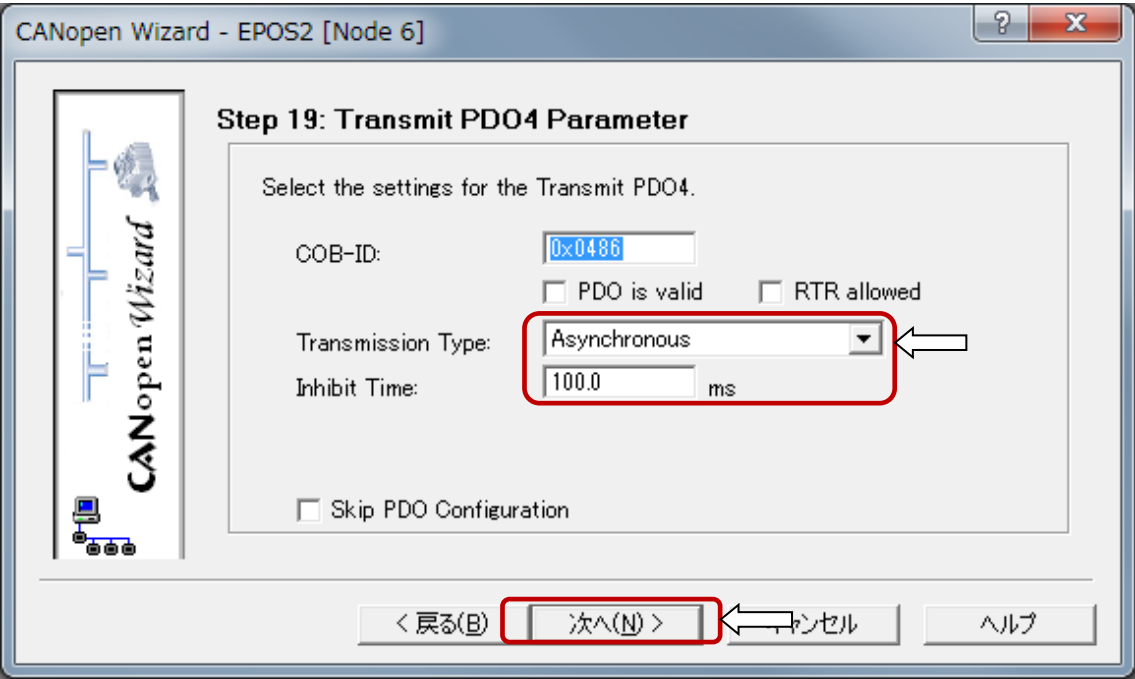


図 47 TPDO Parameter4

TPDO3 のマッピングは Current Actual Value(2Bytes), Encoder Counter(4Bytes)を設定する.

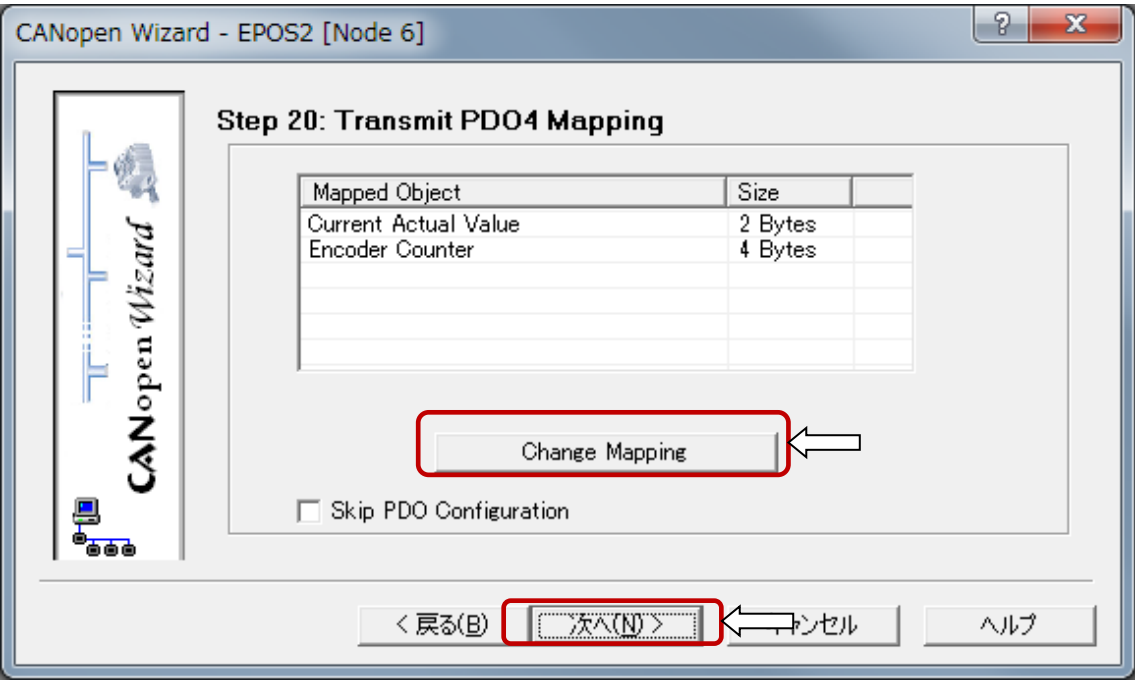


図 48 TPDO mapping

次にハートビート(heartbeat)の時間を設定する。ハートビートとはネットワーク機器が正常に動作しているかを外部に知らせるための信号で、Heartbeat の時間を短くすることで異常を検知するのが早くなる。ここではHeartbeat timeを1000msと設定する。

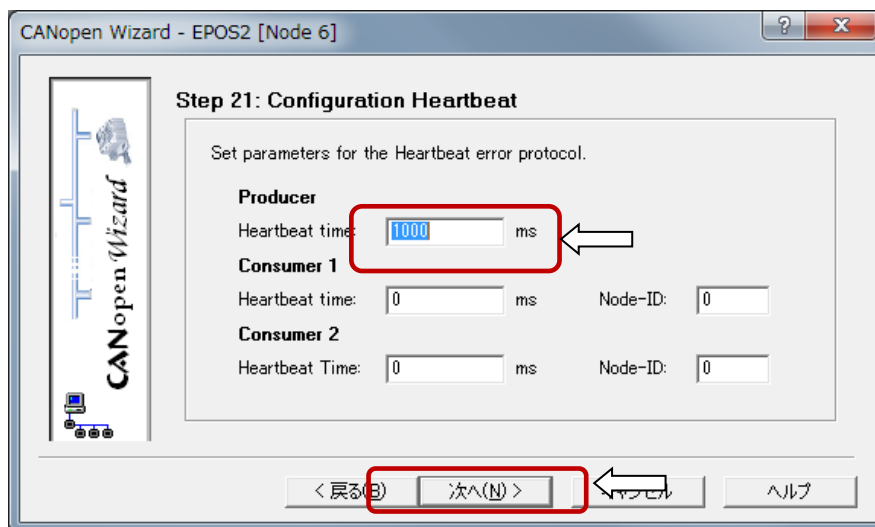


図 49 Heartbeat 設定画面

CANopen が断線したかを判断する Guarding 信号の受信間隔を設定する。(Guarding 受信間隔=Guard time×Life time factor)

設定した時間を超えても Guarding 信号を受信できなかった場合 CANopen が断線したと判断する。

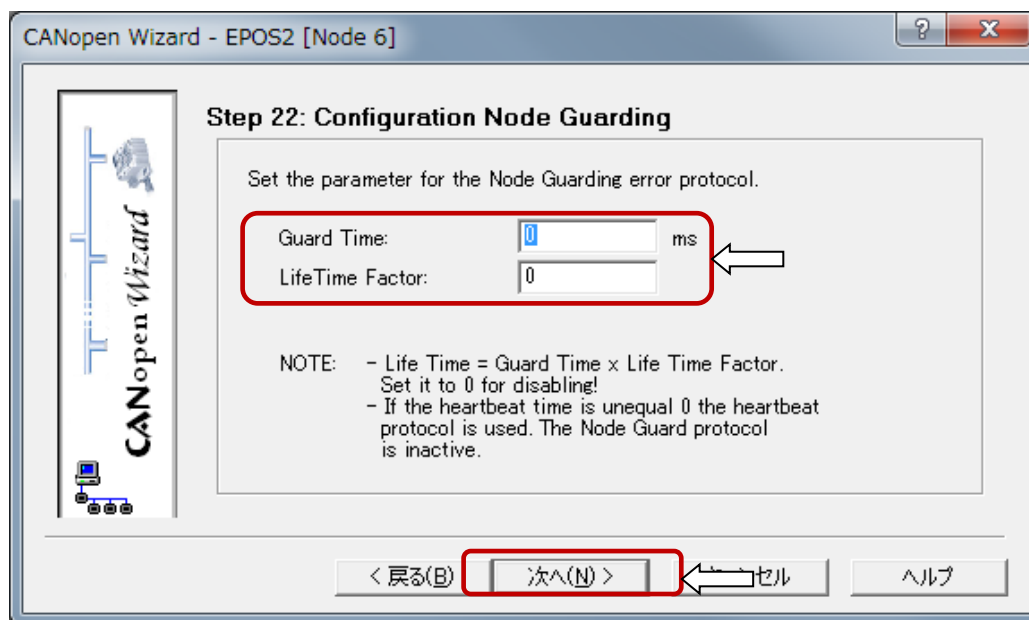


図 50 Configuration Node Guarding 設定画面

設定した値の確認を行う。正しければ「完了」を押す。

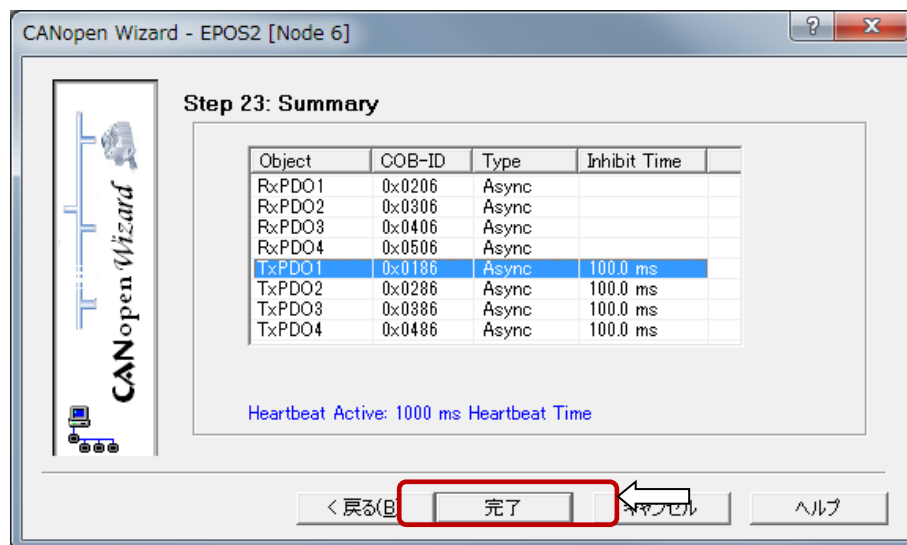


図 51 Guarding Time 設定画面

最後に Navigation ウィンドウの Wizard から Parameter Export/Import を選択する。

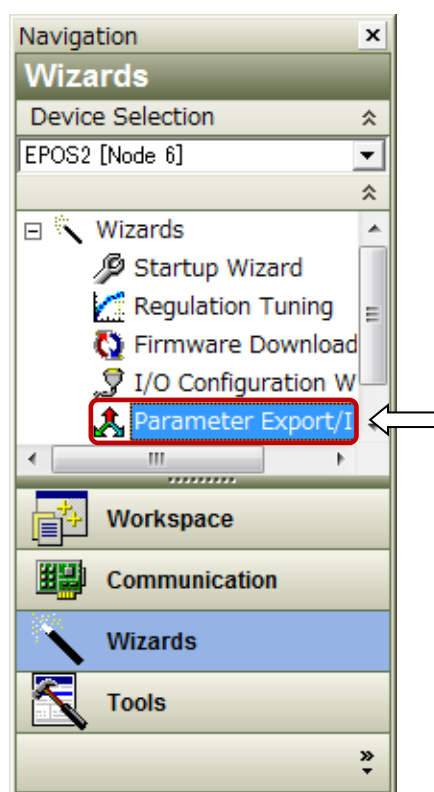


図 52 Parameter Export

DCF を出力するフォルダを選択し, Export Parameters to file をクリックすれば, DCF が生成される. 同様な手順を左モータと右モータの両方に行う.

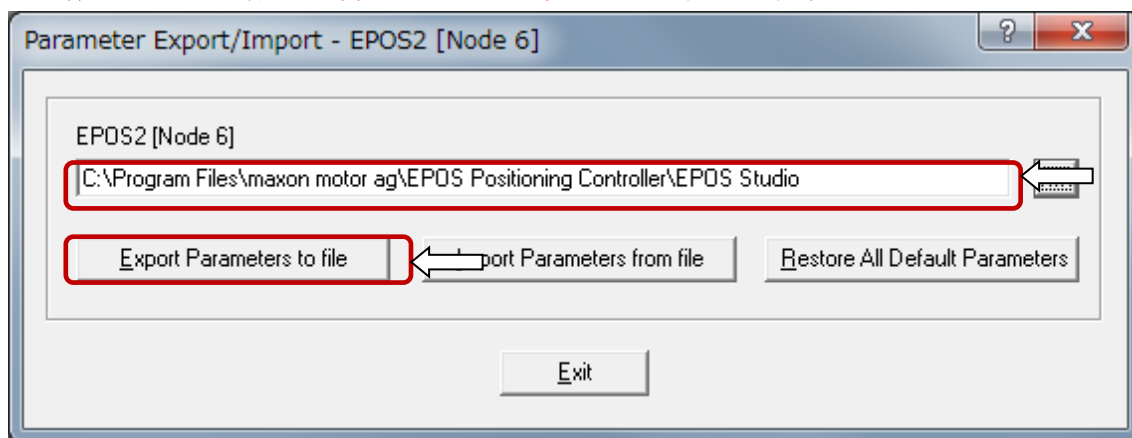


図 53 Parameter Export 設定画面

2.5. Eclipse への RTC-CANopen のツールの導入

ここでは RTC-CANopen の導入方法について説明する。

RTC-CANopen のツール(RTC-CANopen Builder 及び RTC-CANopen SystemEditor)のプラグインを eclipse の plugin のフォルダに入れることで RTC-CANopen のツールの導入が可能となる。これから、RTC-CANopen のツールの利用方法について説明する。

2.5.1. RTC-CANopen Builder のよる DCF の変換及び XML の作成

先ほど生成した DCF を RTC-CANopen を使用するために、DCF の変換及び XML の作成を行う。まず、RTC-CANopen のプラグインを入れた eclipse を立ち上げ、パースペクティブを開く。

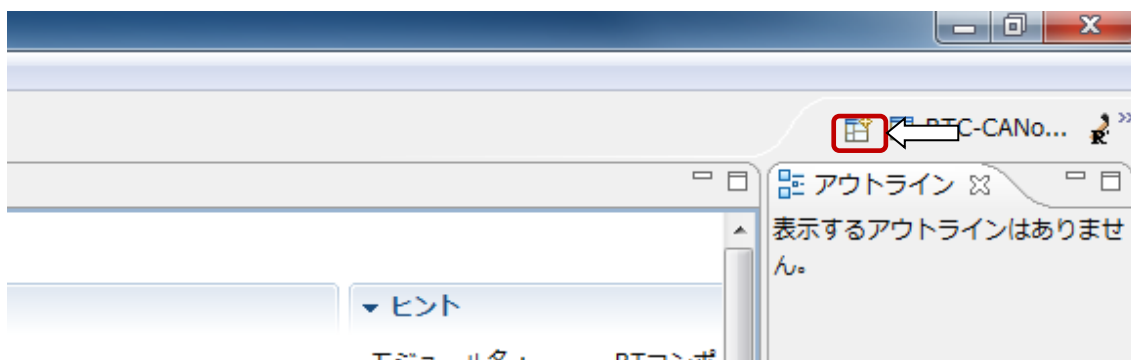


図 54 eclipse パースペクティブ

パースペクティブを開くと以下の画面になるので RTC-Builder を選択し、OK をクリックする。

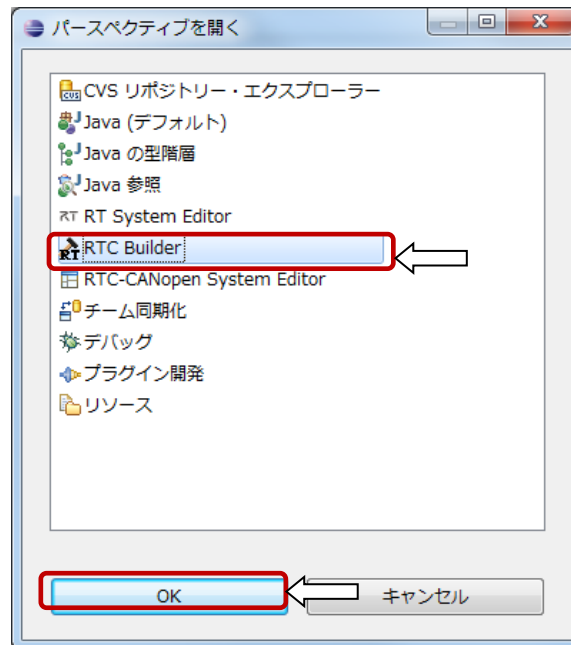


図 55 パースペクティブ設定画面

次にトンカチマークをクリックし、RTC-BUILDER を起動する。



図 56 RTC Builder

プロジェクト名を入力し「完了」をクリックする。

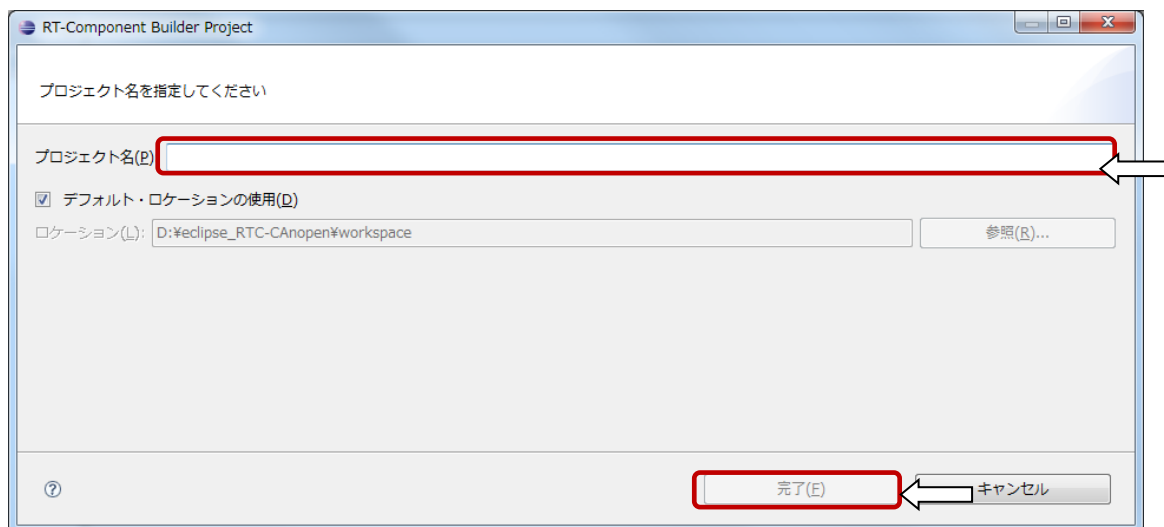


図 57 プロジェクト名指定画面

RTC-Builder を開き、C++(CANopen)を選択する。

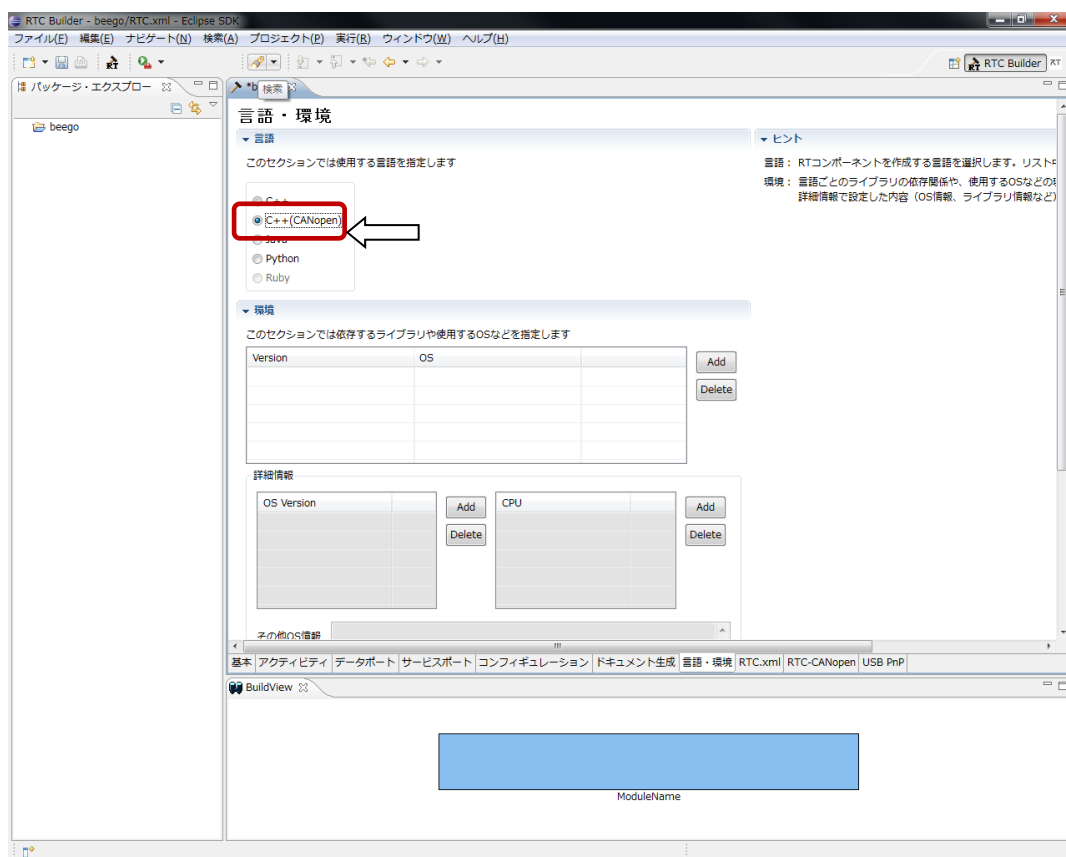


図 58 RTC-Builder 設定画面

次に eclipse の基本のタブをクリックし、CANopen デバイスの DCF を読み込むため、プロファイルの部分の「インポート」を選択する。

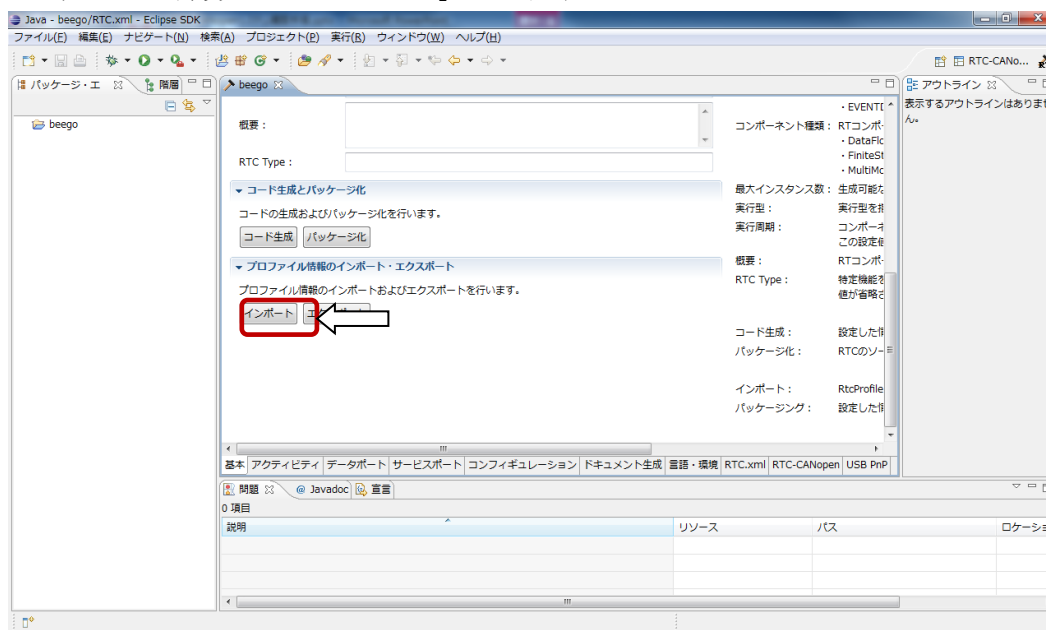


図 59 RTC-Builder 設定画面 2

読み込む DCF のフォルダ(リファレンスロボットでは RightMotor と LeftMotor)まで移動し、DCF を選択したら「開く」を押す。

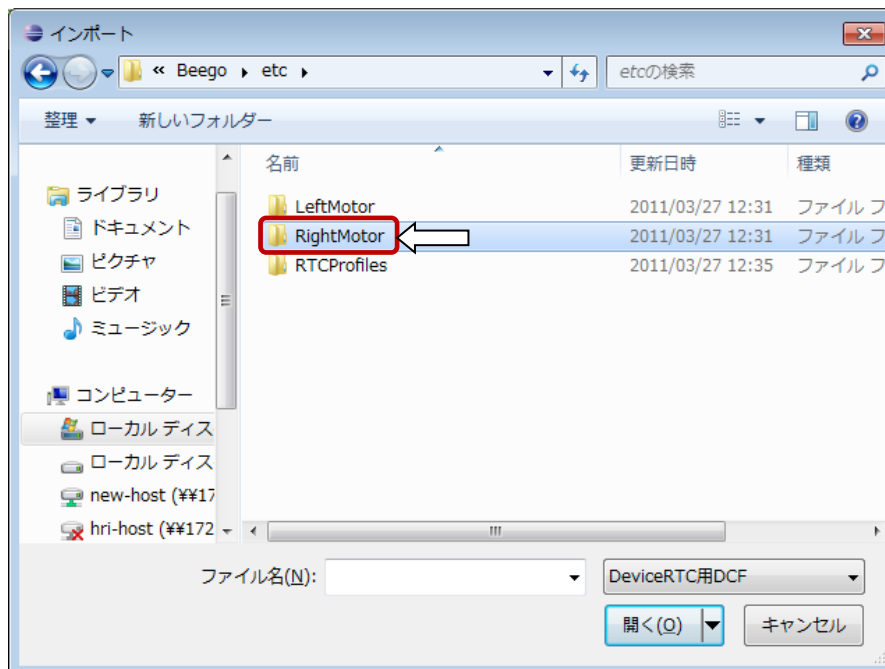


図 60 DCF ファイル指定画面

すると DCF のインポートが完了するので、RTC-CANopen のタブをクリックする。すると以下のような画面になる。この画面ではマッピングパラメータの確認ができ、マッピングが正しいのかを確認する。

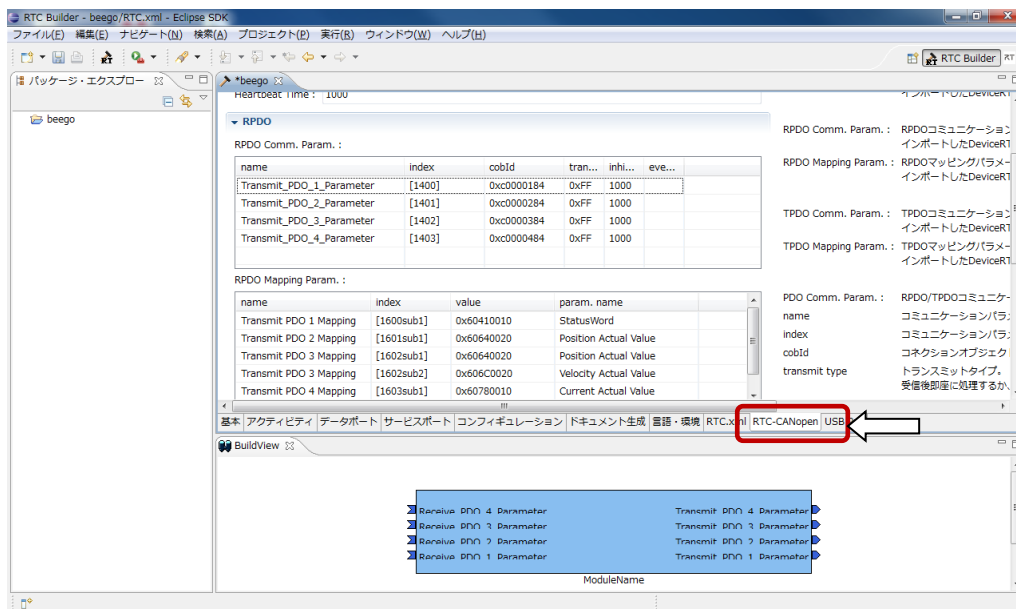


図 61 マッピング情報確認画面

確認が終わったら、RTC Type を 0x01 にする。次に基本のタブからプロファイルの「エクスポート」を選択する。すると ProxyRTC の RTC 仕様が出力される。

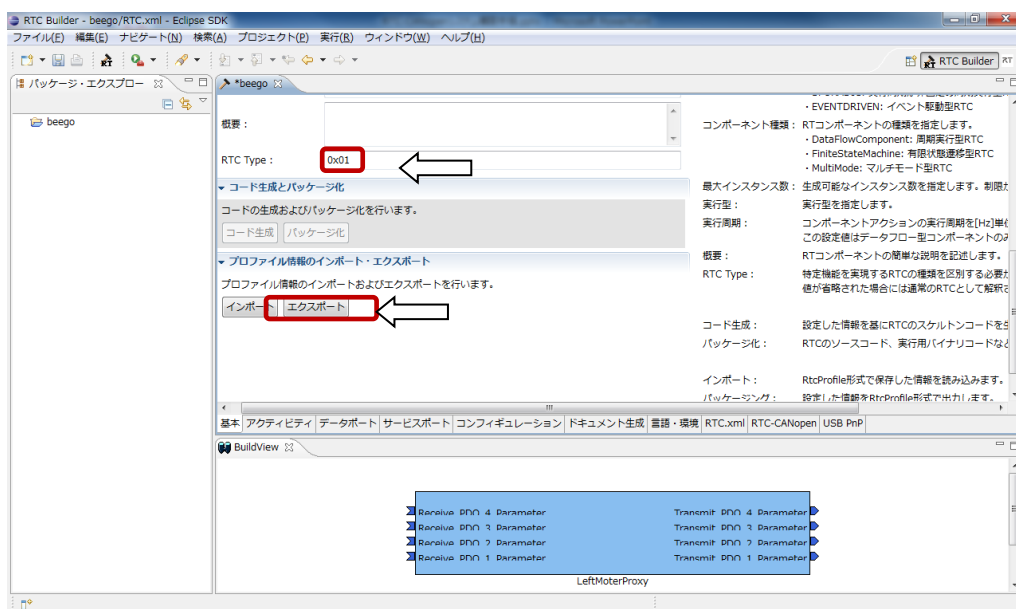


図 62 プロファイル出力画面

2.5.2. RTC-CANopen System Editor による RTS の作成

次に先ほど作成した Proxy 用の DCF と RTC.xml を用いてシステムを構成する。
まず、パースペクティブを開き、RTC-CANopen System Editor を開く。

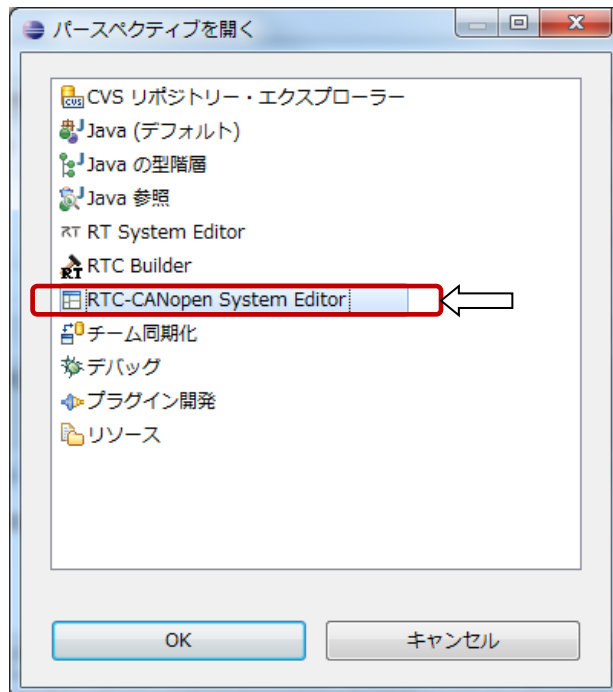


図 63 RTC-CANopen System Editor 選択画面

次に OFF と書かれたボタンを押す。

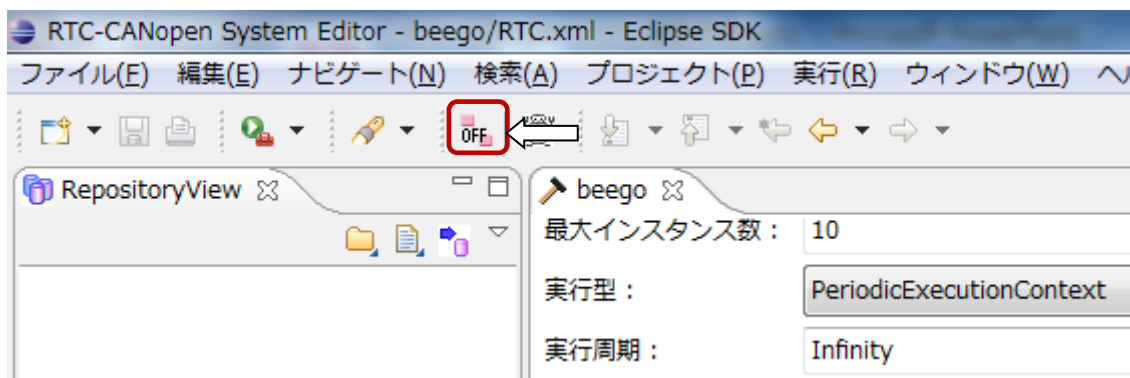


図 64 オフラインエディター選択画面

Device 用 DCF, Proxy 用 DCF, ProxyRTC の RTC 仕様(RTC.xml)を読み込み,
ProxyRTC を作成する. ロボットのマークをクリックする.

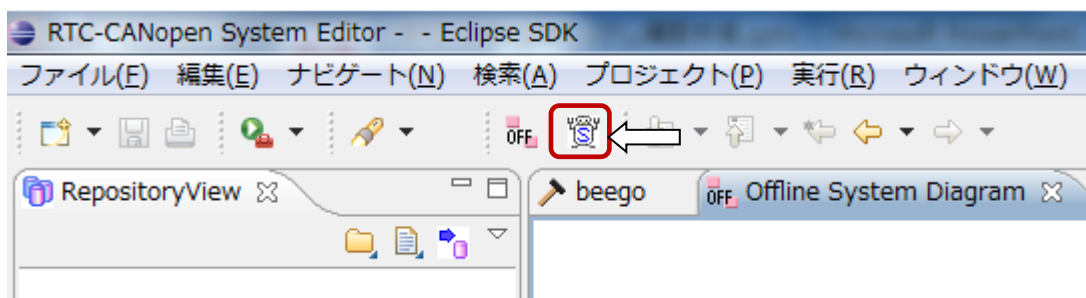


図 65 DCF の読み込み画面

先ほど作成した DCF 及び RTC.xml のフォルダ(リファレンスロボットでは RightMotor
と LeftMotor)を「追加」のボタンで指定してやり, OK を押す.

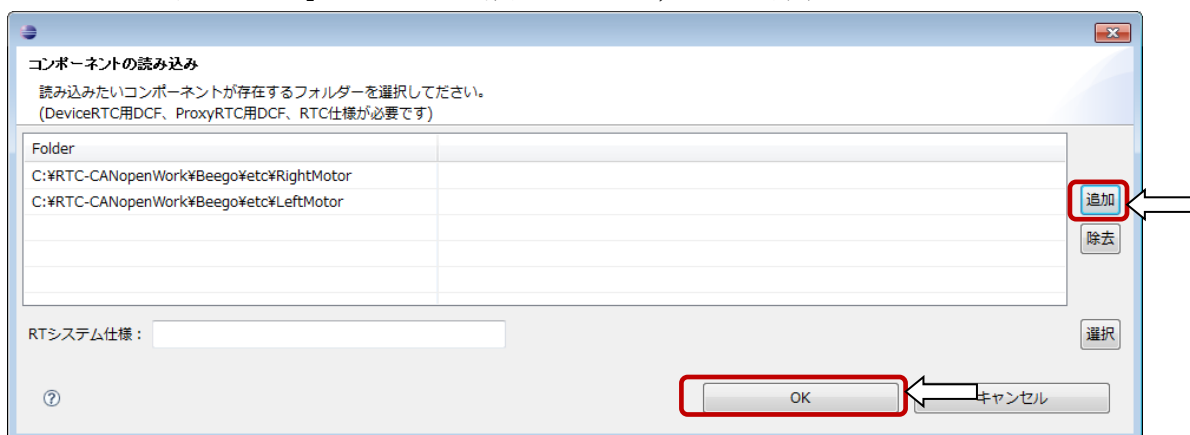


図 66 DCF ファイル選択画面

次に ProxyRTC をコントロールするコンポーネントなどを読み込む。
左の RepositoryView のフォルダを選択する。

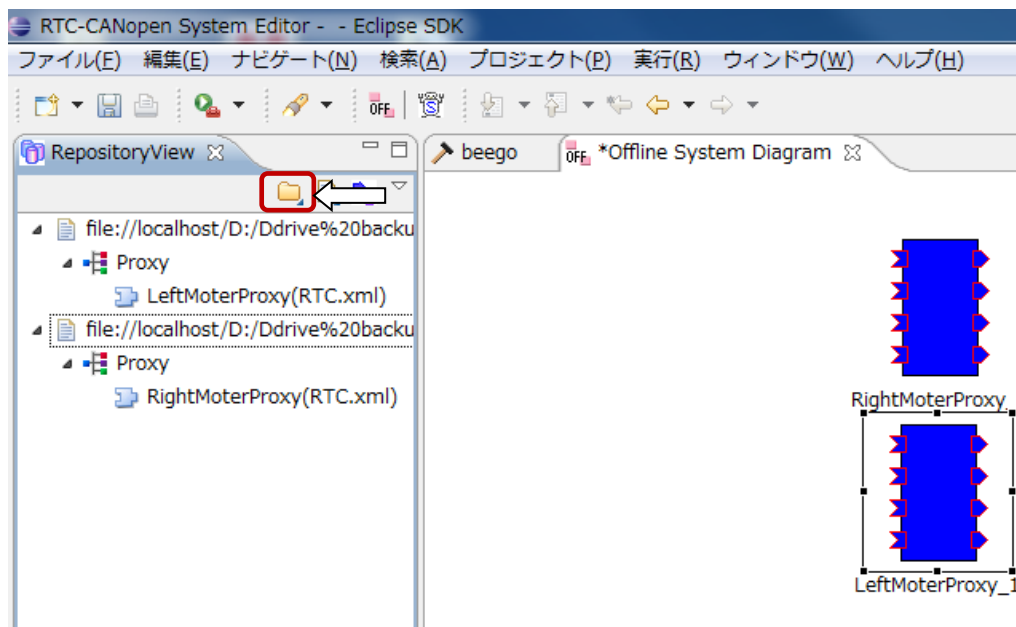


図 67 RTS のフォルダ指定画面

次に構成されるその他のコンポーネントの **rtc.xml**(リファレンスロボットでは **RTCProfiles** 及び **IF_timedVelocity**)を指定してやり、他のコンポーネントを読み込む。
フォルダ指定でコンポーネント読み込むことができるので **rtc.xml** をまとめたフォルダを用意するといいい。

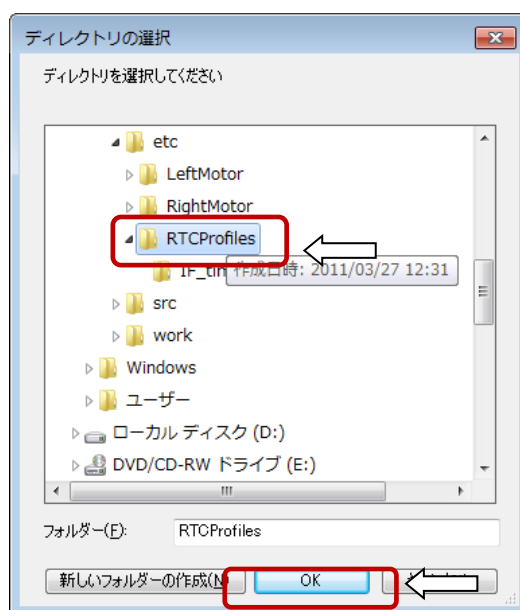


図 68 RTS のディレクトリ指定画面

読み込んだコンポーネントは左の Repository View からドラック & ドロップで表示することができる。これですべてのコンポーネントを表示させる。

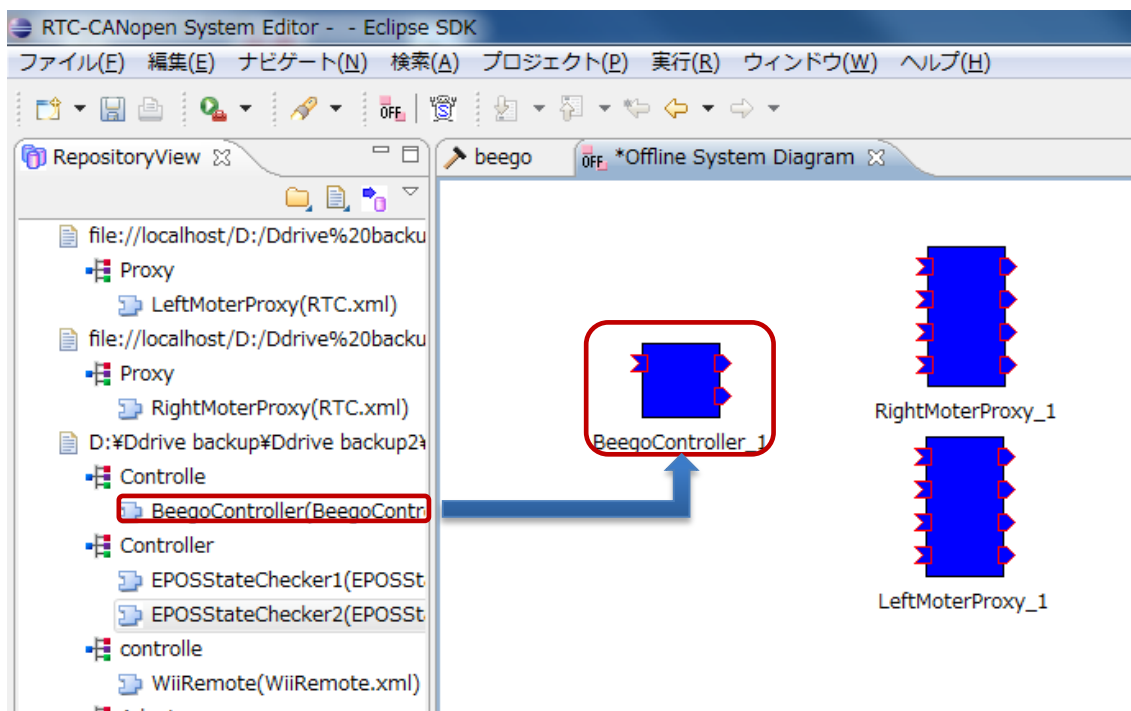


図 69 RTC 選択画面

すべてのコンポーネントを表示させたら、以下のように接続する。

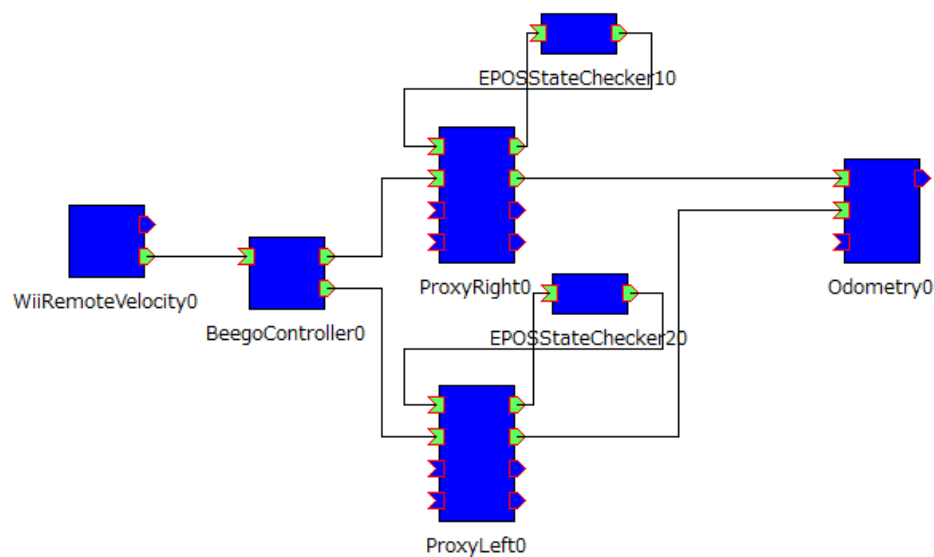


図 70 リファレンスロボット接続例

次に **Offline System Diagram** 内で右クリックし、**Save as** を選択する。

プロファイルの名前を記入する。(すべて半角英語)

Vector : 製作者の名前を記入する。

System Name : システムの名前を記入する。好きな名前を付ける。

Version : システムのバージョンを記入する。

Path : プロファイルをどこに保存するかを決める。「参照」をクリックして保存するフォルダに移動する。

Required : どのプロファイルが必要かを決める。ここでは「すべて選択」をクリックする。

RTC-CANopen 設定 : ProxyRTC の設定を行う。ProxyRTC の実行ファイル(リファレンスロボットでは `bin/gpproxy/GPPProxyRTC.exe`)及び `RTC.conf` ファイル (リファレンスロボットでは `etc/pnpmanager.conf`) を設定する。

すべての記入が終わったら、**OK** をクリックする。

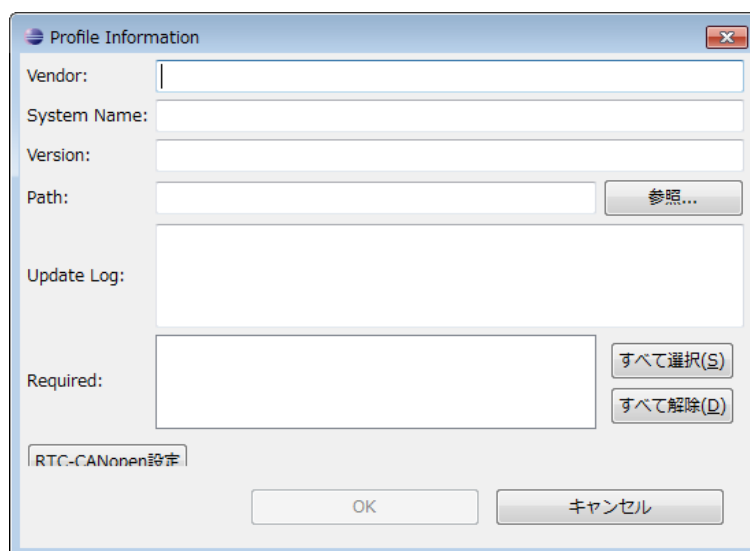
The image shows a Windows-style dialog box titled "Profile Information". It has several input fields: "Vendor:", "System Name:", "Version:", "Path:", and "Update Log:". The "Path:" field has a button labeled "参照..." (Reference...) next to it. Below the "Update Log:" field is a "Required:" section with a list box and two buttons: "すべて選択(S)" (Select All) and "すべて解除(D)" (Deselect All). At the bottom of the dialog, there is a tab labeled "RTC-CANopen設定" and two buttons: "OK" and "キャンセル" (Cancel).

図 71 プロファイル情報記入例

生成されたプロファイルの修正を行う。

1. `instanceName` がデフォルトでは `〇〇_1` (〇〇はコンポーネント名) となっているので `〇〇0` に変更する。

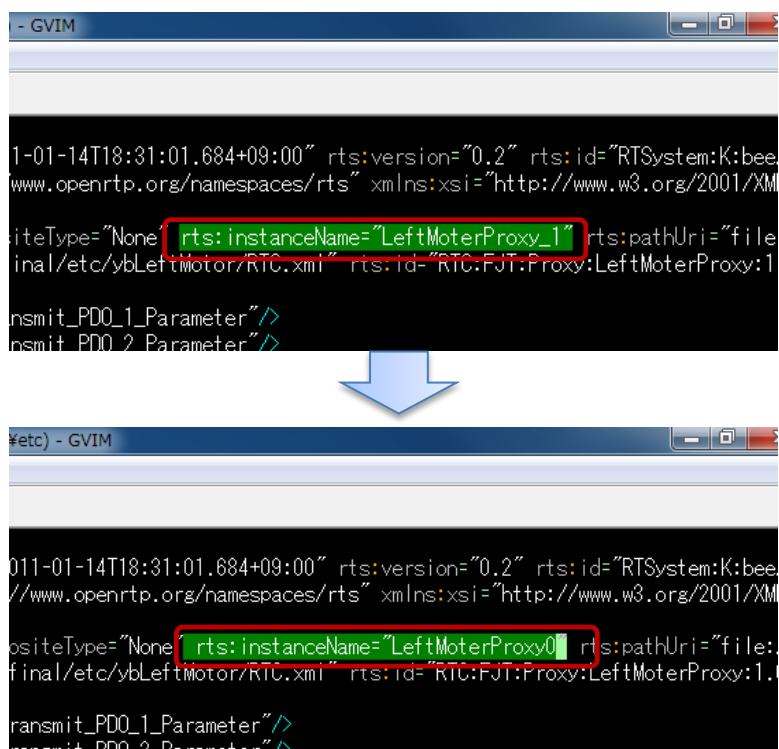


図 72 instanceName 変更例

作成した RTS を実行するため、RTS(リファレンスロボットでは `beego_demo.xml`)までのパスを入力する。

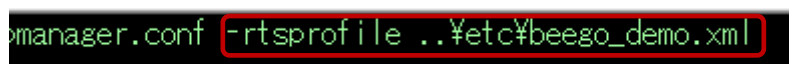


図 73 実行する RTS のパス指定

3. ライセンス

RTC-CANopen リファレンスマニュアルの著作権は, 芝浦工業大学水川研究室に帰属します. これらの RT コンポーネントは, 非営利目的での使用及び改変自由です.

商用利用の場合は別途ご相談下さい.

なお, 本モジュール群を使用して発生したいかなる障害についても責任を負いません.

参考文献

- [1] OpenRTM-aist : <http://www.openrtm.org/>
- [2] ベクタージャパン株式会社 : http://www.vector.com/vj_index_jp.html