

RTC-CANopen リファレンスマニュアル ～RTC-CANopen 詳細説明～

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

目次

1. RTC-CANopen	4
1.1. RTC-CANopen とは?	5
1.2. RT-Middleware とは?	5
1.3. CANopen とは?	5
1.4. RTC-CANopen の機能.....	7
1.5. RTC-CANopen のシステム構成.....	7
1.6. PnP Manager	9
1.7. Status Manager.....	9
1.8. RTC-CANopen Manager.....	10
2. RTC-CANopen を適用したリファレンスロボット.....	11
2.1. 機能.....	11
2.2. ハードウェア構成.....	12
2.3. ソフトウェア構成.....	13
2.4. 構成されているコンポーネントの詳細説明	15
2.4.1. WiiRemoteVelocity.....	15
2.4.2. Beego Controller.....	16
2.4.3. EPOSStateChecker1, EPOSStareChecker2	18
2.4.4. Odometry	20
参考文献	21

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

1. はじめに

本書はRTC-CANopen 及び RTC-CANopen を適用したロボット(リファレンスロボット)の説明をする.

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 構成方法～	RTC-CANopen の実行環境を整える手順が記載してある.

2. RTC-CANopen

2.1. RTC-CANopen とは？

RTC-CANopen とは CANopen という通信規格を RTC にマッピングした RT-Middleware である。RTC-CANopen は CANopen の DCF というデバイス情報が記述されたファイルを読み込むことで Ethernet と CAN のインターフェイス変換を行う ProxyRTC を自動的に生成する。それにより CANopen 製品を容易に RT-Middleware に導入することが可能となる。

2.2. RT-Middleware とは？

現状のロボット開発は企業独自の仕様に元づいて作成されており、再利用性がなく開発効率も悪いため普及させるのが困難となる。

この問題を解決するため、開発されたのが RT-Middleware である。RT-Middleware とはロボットのソフトウェアに共通なインターフェイスを持たせることにより、多様なロボットシステムを簡単に構築することを可能にしたロボット開発用のプラットフォームである。

2.3. CANopen とは？

RTC-CANopen を知る上でもう一つ欠かせないのが CANopen である。CANopen とは耐ノイズ性の強化を考慮して設計され、相互接続された機器間のデータ通信に使われる規格である CAN をベースとした通信プロトコルである。CANopen は産業機器、輸送機器及び工場のオートメーションなどで多く用いられている信頼性のあるバスシステムと言える。さらに CANopen にはノイズに強いという特徴があるため、移動ロボットにも適している。



図 1 CANopen 製品例

また、CANopen は通信層、Object Dictionary、アプリケーション層と分かれている。そのため、通信層をほかの通信規格である Ethernet, FlexRay に変えることが可能となる。

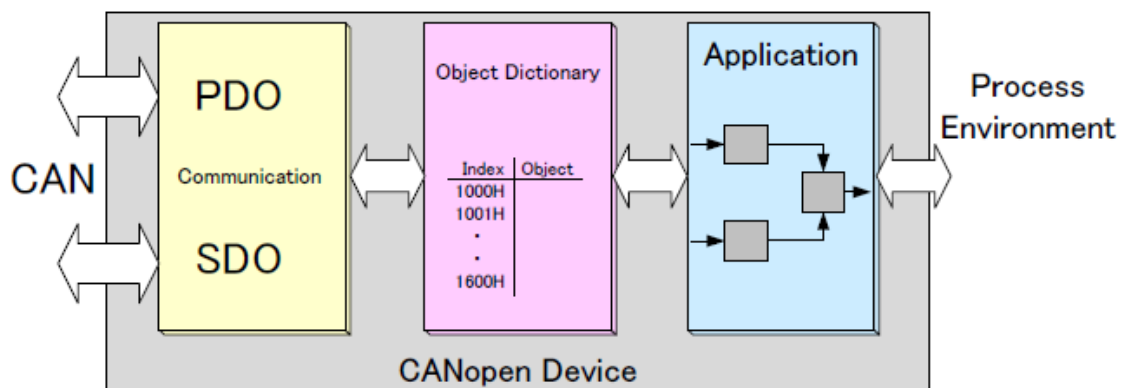


図 2 CANopen デバイスモデル

CAN では同じバスに接続されている各ノード同士で相互通信が可能となっており、CAN メッセージという規格に則り通信を行っている。
また、オブジェクトディクショナリへのアクセスをする通信を SDO、プロセスデータ通信を PDO と言う。

3. RTC-CANopen の機能

前の項で説明した RT-Middleware に CANopen 機能を追加したのが RTC-CANopen となる。RTC-CANopen の機能は以下である。

- CANopen 製品を RT-Component として扱うことができ、ハードウェアの再利用性が向上
- RT-Middleware を用いることができるためソフトウェアの再利用性を向上
- Status Manager により登録されたコンポーネントの監視が可能
- PnP Manager によるコンポーネントの自動接続

3.1. RTC-CANopen のシステム構成

RTC-CANopen のシステムは主に以下の要素からなる。

- Device RTC: CANopen 準拠のデバイスを指す。厳密には RT-Component ではない、以下に示す Proxy RTC により便宜上 RT-Component として扱う。
- Proxy RTC: CANopen デバイスと RT-Component のインターフェイスは異なるため、インターフェイス変換を行う。
- RTC-CANopen Server: PnP Manager, Status Manager, RTC-CANopen Manager のことを指す。

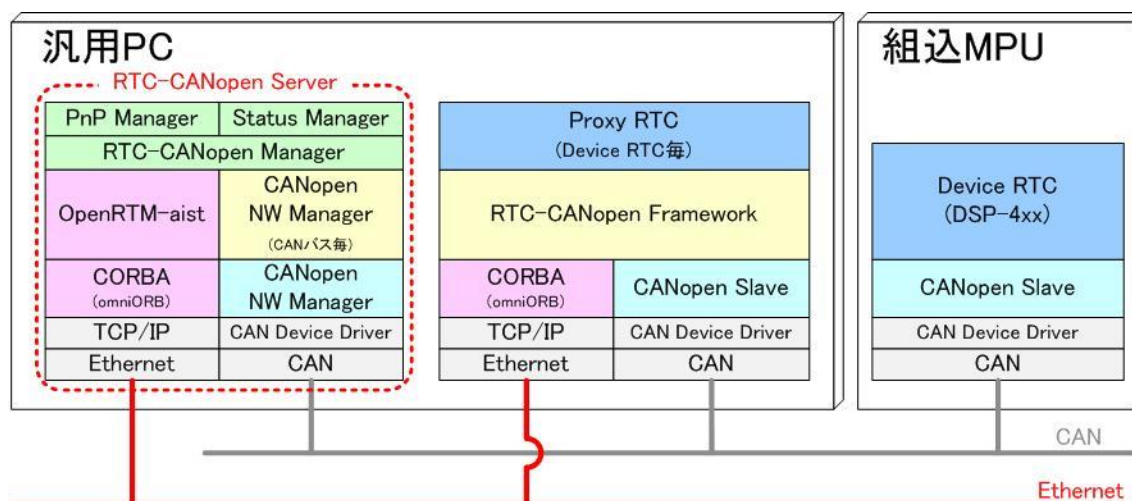


図 3 RTC-CANopen のシステム構成

RTC-CANopen の内部構成と構成するコンポーネントの関係を図 4、図 5 に示す。なお、モデル図はモデリング言語である SysML(Systems Modeling Language)^[1]と UML(Unified Modeling Language)^[2]を用いている。

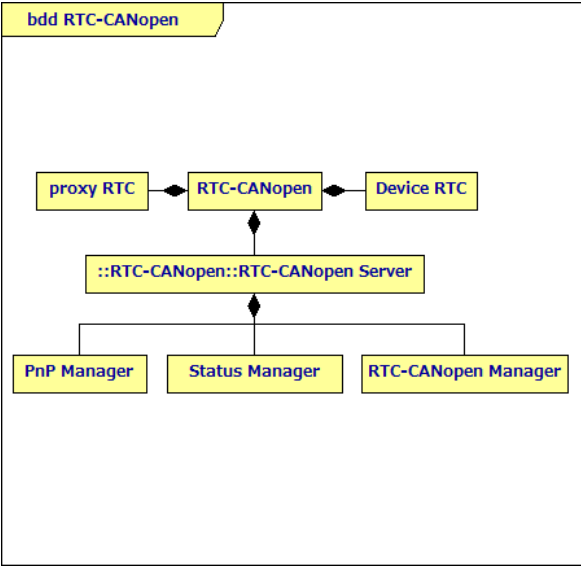


図 4 RTC-CANopen 内部構成

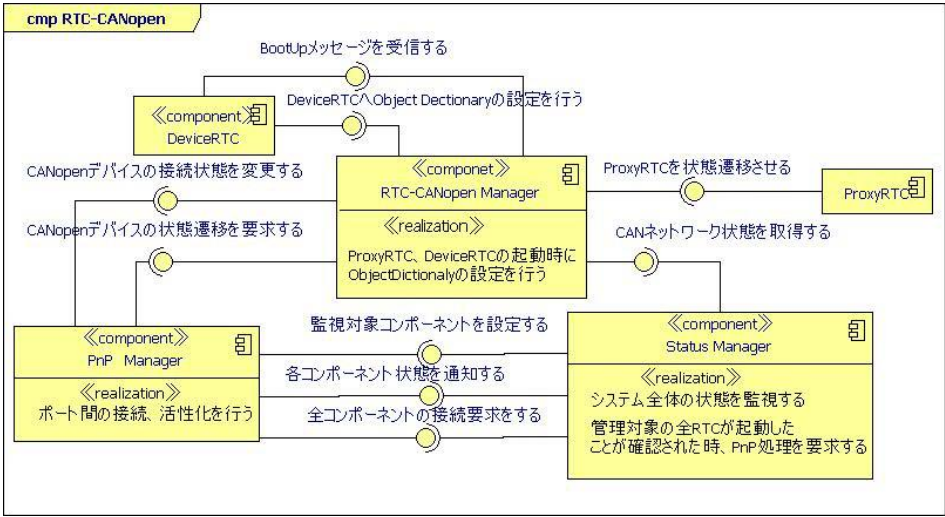


図 5 RTC-CANopen コンポーネント図

PnP Manager, Status Manager, RTC-CANopen Manager については 1.5 で説明する。

3.2. PnP Manager

PnP Manager は、コンポーネントの PnP(プラグアンドプレイ)機能を実現するモジュールであり、システム全体構成の構築、管理を行う。主な働きは以下となる。

- ロボットシステム仕様記述(RTS)を読み込み、RTC-CANopen Manager に対し、サービスポートを通じて設定を行う。
- DeviceRTC 起動を RTC-CANopen Manager から通知を受け、ProxyRTC の起動を指示する。
- ProxyRTC を含む RTC の状態変化があった場合、Status Manager から通知を受ける。
- システム内の全 RTC が起動したタイミングで、RTC のポート間接続、活性化を行う。

3.3. Status Manager

Status Manager は、システム上で動作する RTC のステータスデータを収集し、システム全体の状態監視を行う。

- RTC の状態監視を行う。
- デバイスの状態監視を行う。
- CAN ネットワークの監視を行う。

3.4. RTC-CANopen Manager

RTC-CANopen Manager は, OpenRTM-aist のフレームワークと CANopen のフレームワークの両方を持ち, 上位アプリケーション(PnP Manager, 及び Status Manager) とフレームワークとのインターフェイス変換を行うモジュールである. RTC-CANopen 上で発生したイベントを上位アプリケーションへ通知する機能, 及び上位アプリケーションからの要求に応じて, ProxyRTC, DeviceRTC に対して動作要求を行う機能を持つ.

- PnP Manager, Status Manager より動作要求を受け, ProxyRTC, DeviceRTC に対して動作要求を行う.
 - DeviceRTC の Bootup メッセージを受信したタイミングで, ProxyRTC のプロセスを起動する.
 - ProxyRTC, DeviceRTC の起動時に, CANopen のメッセージ通信により ObjectDictionary の設定を行う.
 - ProxyRTC と DeviceRTC の状態が同期するように状態遷移を行う.
- 各種異常状態を検出する.
 - ロボットシステム仕様記述で定義していないデバイスをシステムに接続した場合, 検出することができる. 不正デバイス, または不正 ID を検出した場合は PnP Manager に対して異常通知を行う.
 - DeviceRTC からの Heartbeat が規定時間内に来ない場合, DeviceRTC の異常とみなし, PnP Manager に通知する.

4. RTC-CANopen を適用したリファレンスロボット

4.1. 機能

RTC-CANopen を利用するために，RTC-CANopen を適用したロボットがリファレンスロボットである．リファレンスロボットの動作はとてもシンプルで，リモートコントローラによりロボットを手動で操作する(ラジコンのような動作)．また，速度と角速度を入力することにより，ロボットの自己位置(x , y , θ)が出力されるシステムとなっており，ほかのコンポーネントとの組み合わせも考慮されている．インターフェイスは次世代ロボット知能化技術開発プロジェクトの移動知能第一サブワーキンググループで統一された仕様に基づいており，再利用性の高いものとなっている．以下の図はリファレンスロボットの提供する機能をユースケース図で示したものになる．

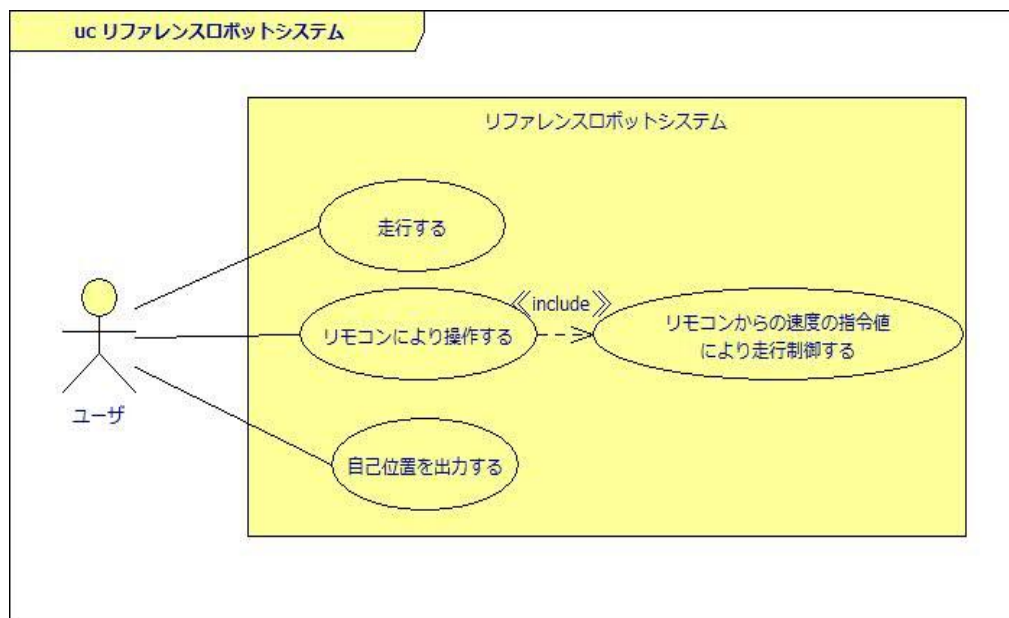


図 6 リファレンスロボットユースケース図

4.2. ハードウェア構成

はじめに、ハードウェアの構成について説明する。

リファレンスロボットのハードウェアの構成要素をモデル化したものは以下の通りである。ハードウェア構成は主にエンコーダ付きモータ、EPOS, beego, PC, CAN ネットワークインターフェイス(CANcardXL)からなる。

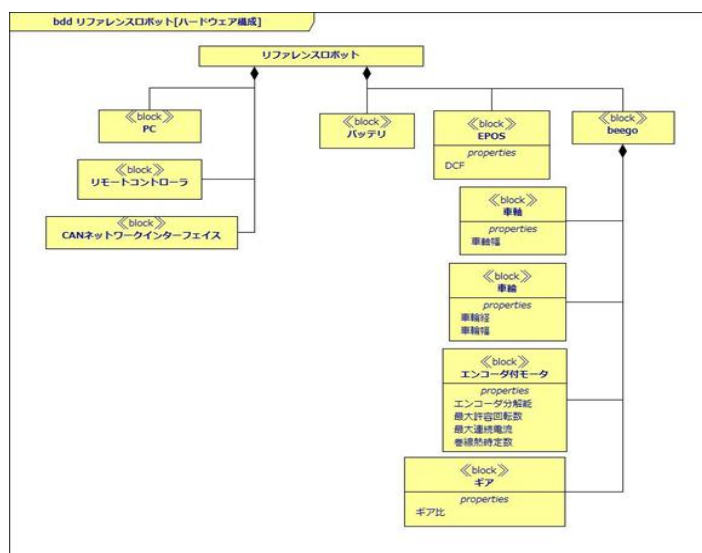


図 7 リファレンスロボットのハードウェア構成(ブロック図)

ハードウェア要素の関係を表したものが下の図になる。PC と CAN ネットワークインターフェイス(CANcardXL)は Ethernet で情報をやり取りし、CANcardXL と EPOS は CANopen で通信する。

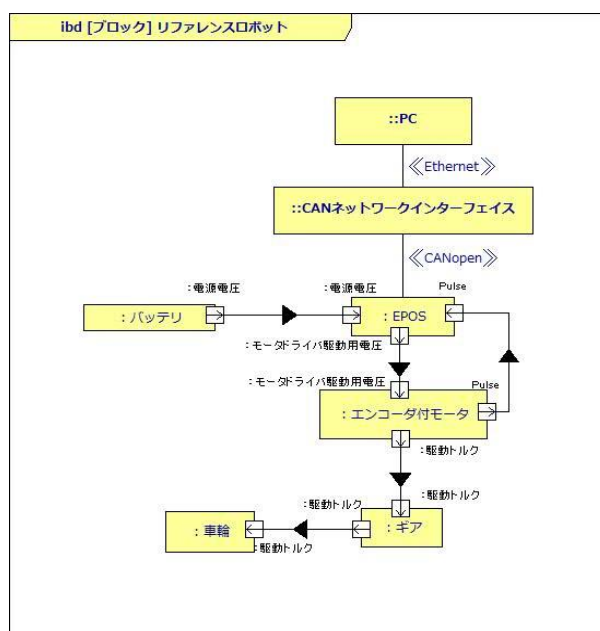


図 8 リファレンスロボットハードウェア構成

4.3. ソフトウェア構成

次にソフトウェアの構成について説明する．ソフトウェアの構成は主に以下で構成される．

1. WiiRemoteVelocity : Wii リモコン(クラシックコントローラ付属)の入力を速度と角速度に変換する．
2. BeegoController : CANopen デバイス EPOS に指令値を与える．
3. ProxyRight, ProxyLeft : Ethernet と CAN のインターフェイス変換を行う ProxyRTC となる．
4. EPOSStateChecker1, EPOSStateChecker2 : EPOS の状態遷移を行う．
5. Odometry : ProxyRTC からのエンコーダカウントにより自己位置(x. y. θ)を出力する．

コンポーネントの詳しい説明は 4.4 を参照のこと．

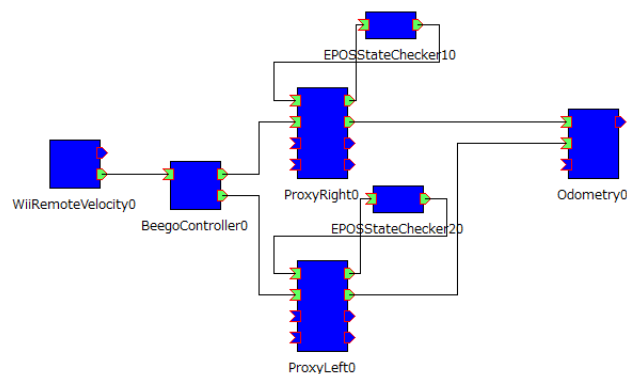


図 9 リファレンスロボットコンポーネント接続図

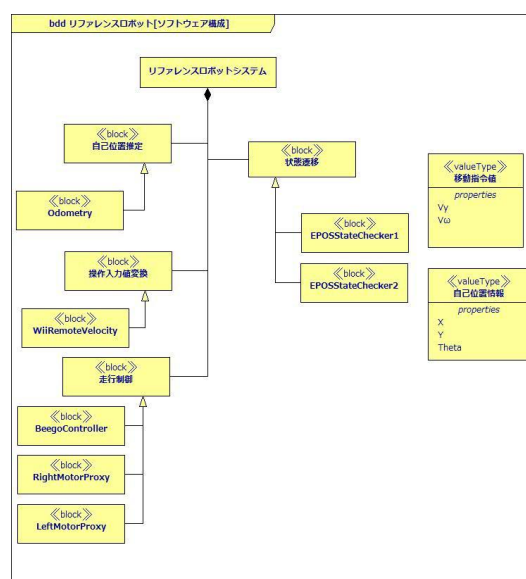


図 10 リファレンスロボットのソフトウェア構成(ブロック図)

構成されるコンポーネントの関係を記した内部ブロック図が以下となる。

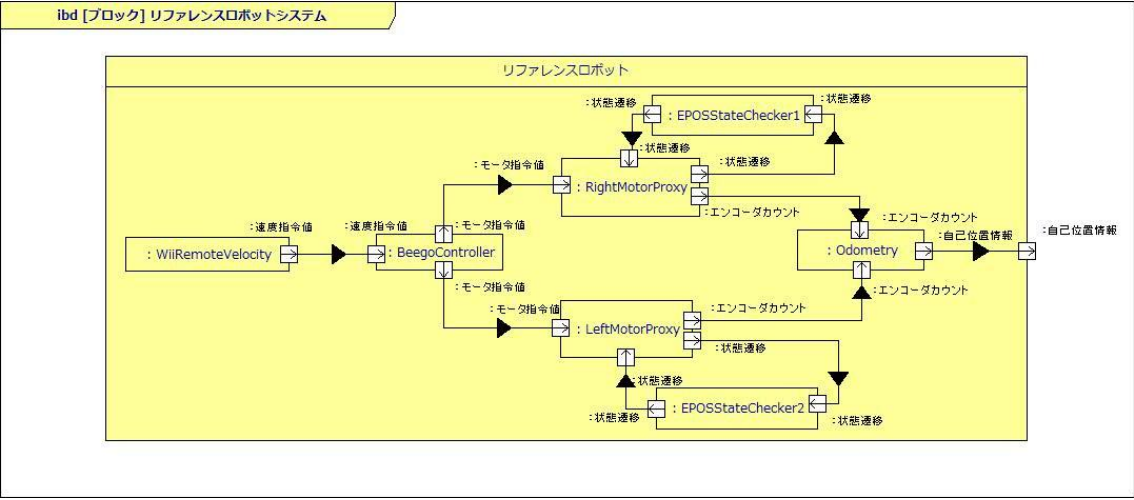


図 11 リファレンスロボットのソフトウェア構成(内部ブロック図)

使用しているモータドライバ EPOS の設定及びマッピングについて説明する。
EPOS の設定については

http://www.maxonjapan.co.jp/manual/epos/EPOS_Application_Note_DeviceProgramming.pdf の EPOS Application Note デバイスプログラミングを参照にする。
今回のリファレンスロボットは Profile Velocity Mode を使用する。

リファレンスロボットのモータドライバ EPOS の TPDO 及び RPDO のマッピングを示す。

表 2 リファレンスロボットのマッピング

PDO type	Mapping Parameter[Byte]							
	0	1	2	3	4	5	6	7
RPDO 0	ControlWord							
RPDO 1	ControlWord		Modesof Operation	TargetVelocity				
RPDO 2	ControlWord		Modesof Operation	TargetPosition				
RPDO 3	ControlWord		Modesof Operation	TargetVelocity				
TPDO 0	StatusWord		TargetVelocity					
TPDO 1	PositionActualValue							
TPDO 2	PositionActualValue				VelocityActualValue			
TPDO 3	CurrentActualValue		Encoder Counter					

4.4. 構成されているコンポーネントの詳細説明

次にコンポーネント単位で説明する.

4.4.1. WiiRemoteVelocity

コンポーネント概要 : Wii リモコン(クラシックコントローラ付属)からの入力を速度と角速度に変換する.

- 動作環境

動作 OS	Windows XP Windows 7
動作 RTM	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
依存ライブラリ	なし

- 動作条件

動作周期	50Hz
------	------

- 出力ポート

データポート

名称	データ型	データ長	説明
TargetVelocity	IIS::TimedVelocity2D	—	速度出力(速度, 角速度)

4.4.2. Beego Controller

コンポーネント概要：入力される速度，角速度を CANopen デバイス EPOS に伝達する．

- 動作環境

動作 OS	Windows XP Windows 7
動作 RTM	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
依存ライブラリ	なし

- 動作条件

動作周期	50Hz
------	------

- 入力ポート

データポート

名称	データ型	データ長	説明
InputTargetVelocity	IIS::TimedVelocity2D	－	EPOS を動かすための速度(速度，角速度)を入力する．

● 出力ポート
データポート

名称	データ型	データ長	説明
TxPDOMsgRight	RTC::TimedOctetSeq	8	<p>必要なデータは EPOS の DCF を参照にする.</p> <p>Control Word や modes of Operation の意味は EPOS の説明書を参照にする.</p> <p>配列 No. : [0~1] Control Word = 0x0F</p> <p>配列 No. : [2] Modes of Operation = 3 (Profile Velocity Mode)</p> <p>配列 No. : [3~6] 目標回転数</p> <p>配列 No. : [7] reserved</p>
TxPDOMsgLeft	RTC::TimedOctetSeq	8	<p>必要なデータは EPOS の DCF を参照にする.</p> <p>Control Word や modes of Operation の意味は EPOS の説明書を参照にする.</p> <p>配列 No. : [0~1] Control Word = 0x0F</p> <p>配列 No. : [2] Modes of Operation = 3 (Profile Velocity Mode)</p> <p>配列 No. : [3~6] 目標回転数</p> <p>配列 No. : [7] reserved</p>

4.4.3. EPOSStateChecker1, EPOSStareChecker2

コンポーネント概要：Maxon 社製モータドライバである EPOS 内部の状態遷移を行う。状態遷移させる対象は EPOS がモータドライバとして持っている状態遷移であるので、CANopen の状態遷移ではないので注意が必要。EPOS 内の状態については、EPOS の Firmware マニュアルを参照のこと。主に switch on disable～operation enable の状態遷移と Fault～operation enable までの状態遷移を行う。

- 動作環境

動作 OS	Windows XP Windows 7
動作 RTM	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
依存ライブラリ	なし

- 動作条件

動作周期	50Hz
------	------

- 入力ポート

データポート

名称	データ型	データ長	説明
StatusWord	RTC::TimedOctetSeq	8	EPOS の現在の状態(Status Word)を入力する。 EPOS のマッピングパラメータが違う場合はソースを編集する必要がある。 マッピングパラメータは DCF を参照にする。 配列 No. : [0～1] Status Word 配列 No. : [2～7] reserved

- 出力ポート

データポート

名称	データ型	データ長	説明
ControlWord	RTC::TimedOctetSeq	8	EPOS の状態遷移を行うコマンドを出力する。

			<p>EPOS のマッピングパラメータが違う場合はソースを編集する必要がある.</p> <p>マッピングパラメータは DCF を参照にする.</p> <p>配列 No. : [0] Control Word</p> <p>配列 No. : [1～7] reserved</p>
--	--	--	--

4.4.4. Odometry

コンポーネント概要：モータからの出力値(エンコーダカウント)から自己位置を算出する。

動作環境

動作 OS	Windows XP Windows 7
動作 RTM	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
依存ライブラリ	なし

● 動作条件

動作周期	50Hz
------	------

● 入力ポート

データポート

名称	データ型	データ長	説明
PositionActualValueLeft	RTC::TimedOctetSeq	8	EPOS からのデータ(エンコーダカウント)を取得する.
PositionActualValueRight	RTC::TimedOctetSeq	8	EPOS からのデータ(エンコーダカウント)を取得する.

● 出力ポート

データポート

名称	データ型	データ長	説明
result Localization	IIS::TimedVelocity2D	—	現在の自己位置を出力する. (単位はメートル)

5. 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] OMG SysML <http://www.object-report.jp/omginfo/technology/sysml/>
- [2] OMG UML <http://www.object-report.jp/omginfo/technology/sysml/>