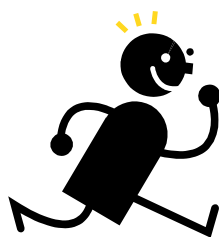


人追尾カメラモジュール制御 RTC ユーザーマニュアル (v1.0)

- ・ モータドライバ RTC
- ・ URG 計測 RTC
- ・ 人追尾コントローラ RTC



2012 年 12 月 7 日

芝浦工業大学 機械機能工学科知能機械システム研究室

(担当：生田目祥吾，石田真一)

～～目次～～

1. はじめに

| | | |
|------|----------------------|-----|
| 1.1. | この RTC で出来ること | 2～3 |
| 1.2. | 動作環境 | 4 |
| 1.3. | OpenRTM-aist インストール | 4 |
| 1.4. | ネーミングサーバの起動 | 5 |
| 1.5. | RT System Editor の起動 | 6 |

2. モータドライバ RTC

| | | |
|------|---------------------|-------|
| 2.1. | モータドライバ RTC の概要 | 7 |
| 2.2. | 使用するハードウェア | 8 |
| 2.3. | 設定用ソフトウェアのダウンロード | 9～11 |
| 2.4. | EPOS2 24/2 の配線と初期設定 | 12～15 |
| 2.5. | RT ミドルウェアを利用した動作確認 | 16～22 |

3. URG 計測 RTC

| | | |
|------|--------------------|-------|
| 3.1. | URG 計測 RTC の概要 | 23 |
| 3.2. | 使用するハードウェア | 24 |
| 3.3. | デバイスドライバのダウンロード | 25～30 |
| 3.4. | RT ミドルウェアを利用した動作確認 | 31～34 |

4. 人追尾コントローラ RTC

| | | |
|------|---------------------|-------|
| 4.1. | 人追尾コントローラ RTC の概要 | 35 |
| 4.2. | RTC ミドルウェアを利用した動作確認 | 36～39 |

5. 人追尾カメラモジュール

| | | |
|------|-------------|-------|
| 5.1. | システム構成 | 40 |
| 5.2. | 既存 RTC の再利用 | 41～43 |
| 5.3. | RTC の接続と実行 | 44 |
| 5.4. | さらに発展した利用法 | 44 |

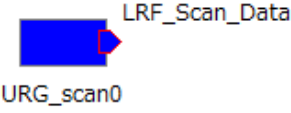
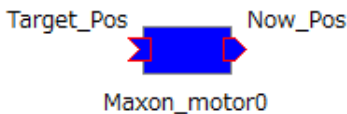
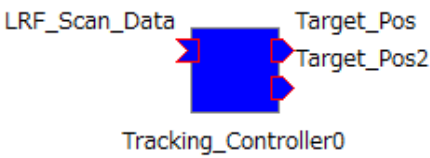
| | |
|---------------------|----|
| 再利用における注意事項（ライセンス等） | 45 |
|---------------------|----|

| | |
|------------|----|
| 参考文献・プログラム | 45 |
|------------|----|

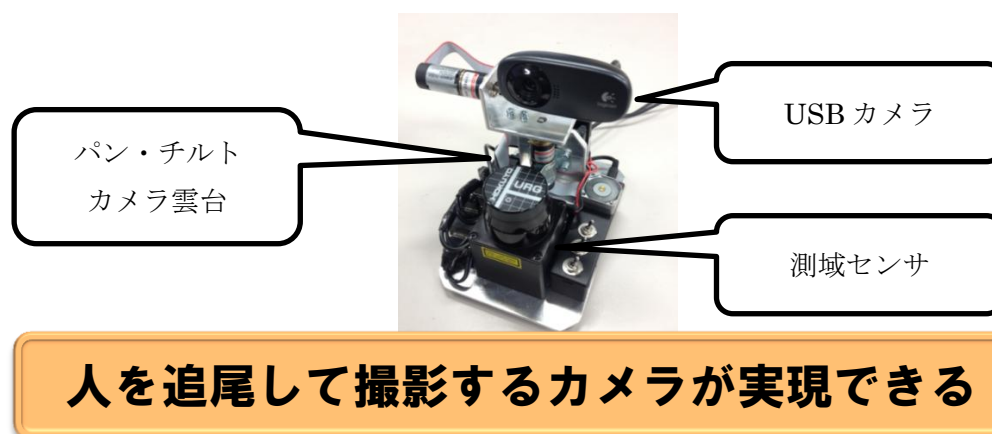
1. はじめに

1.1. この RTC で出来ること

人追尾カメラモジュール制御 RTC は、以下に示す 3 種類の RT コンポーネントです。すべて修正 BSD ライセンスを適用します。なお「モータドライバ RTC」ではマクソンジャパン(株)が、「URG 計測 RTC」では北陽電機(株)がそれぞれ公開しているサンプルプログラムの一部とライブラリ利用しています。

| | | |
|--------------------------|--|--|
| モータドライバ RTC |  | マクソンジャパン(株)で販売されているモータドライバ EPOS2 24/2 を制御する RTC. |
| URG 計測 RTC |  | 北陽電機(株)で販売されている測域センサ URG シリーズからデータを取得する RTC. |
| 人追尾コントローラ RTC |  | 測域センサの計測値から人の存在箇所を推定し、追尾動作指示をモータに出力する RTC. |

これらの RTC を用いて、搭載したカメラが人の動きに合わせて追尾して撮影するように駆動するパンチルトカメラ雲台が実現可能です。下図は作成した追尾カメラモジュールの動作の様子ですが、フレームはホームセンターで購入した金属板を金鋸とハンドドリルで加工したものですので、モータや測域センサさえあれば簡単に作成できます。



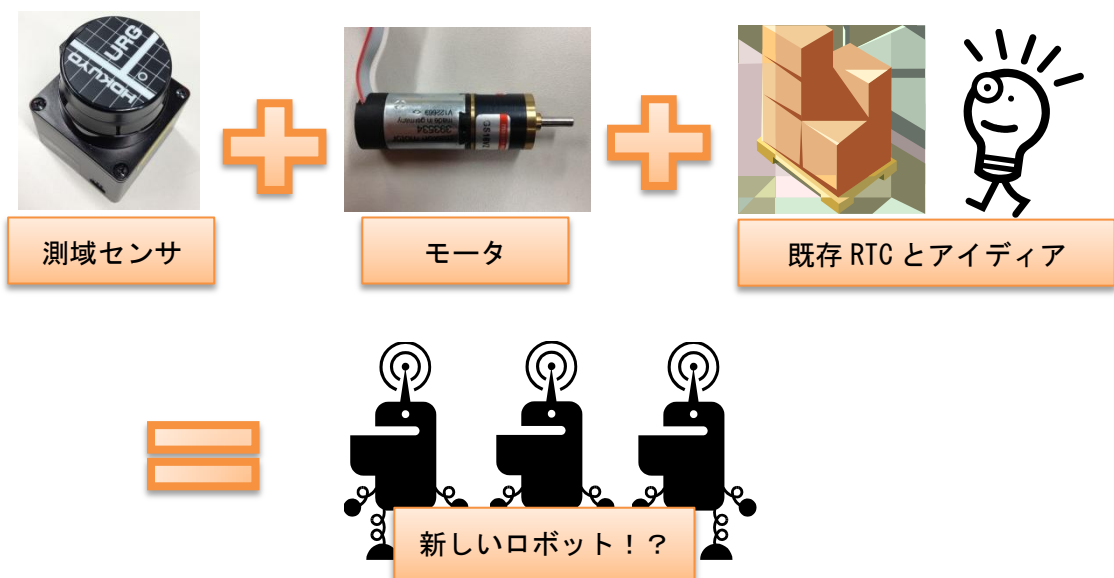
追尾カメラを使えば・・・

追尾カメラは様々な用途で利用することができます。環境に設置することで防犯カメラとして機能しますし、高度な顔認識を搭載することで受付システムとしても利用できるかもしれません。さらに、ロボットに搭載すれば人の存在や環境を認識するセンサとして利用することもできます。



追尾カメラ以外にも・・・

「モータドライバ RTC」と「URG 計測 RTC」は市販の製品を制御するための RTC です。したがって、該当する製品さえ手に入れば誰でもある種の「アクチュエータ」と「センサ」が機能します。あとは既存の RTC と新しいアイデアを上手く組み合わせ、別の「コントローラ」に置き換えることで異なるシステムを構築することもできます。測域センサの計測情報を利用して知的に動作するロボットを色々と考えてみましょう。



1.2. 動作環境

対応している OS は Windows のみです。動作の確認には 64bit 版の Windows7 Home premium および Enterprise 上に C++版の OpenRTM-aist-1.1.0 をインストールした PC を使用しました。開発環境は Visual C++ 2010 Professional を使用しています。



～～動作を確認した PC～～
64bit 版の Windows7
C++版の OpenRTM-aist-1.1.0

1.3. OpenRTM-aist のインストール

RT ミドルウェアに全く触ったこともないという人もいると思いますので、OpenRTM-aist をインストールする方法を説明します。既にインストールされている場合は以下を読み飛ばしてしまいましょう。

32bit Windows 環境は

OpenRTM-aist は各 RTC を動作させる上で必要不可欠ですので必ずインストールしなくてはなりません。バージョンは OpenRTM-aist1.1.0 というものを利用します。ダウンロードは以下の URL が示す OpenRTM-aist 公式 HP から行うことができます。32bit 版 Windows 環境の場合は上記 HP に詳しい解説が載っていますので非常に簡単に設定ができます。

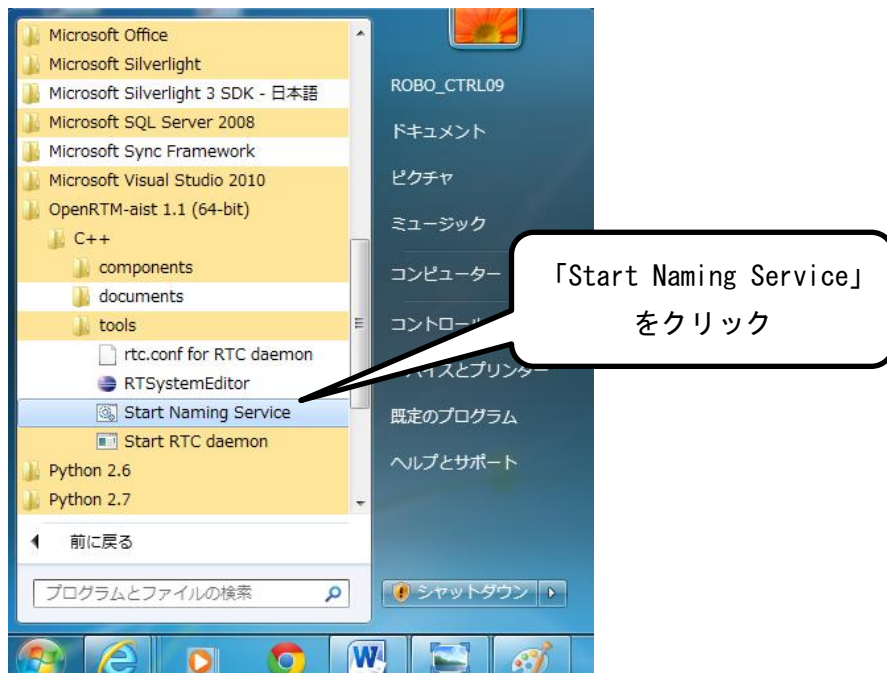
<http://www.openrtm.org/openrtm/ja>

64bit Windows 環境は

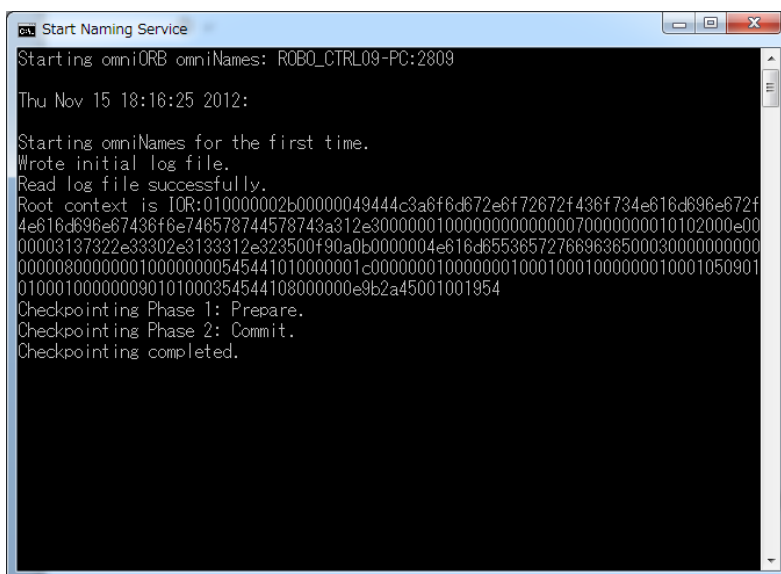
最近 64bit の OS で動いている PC が随分と増えてきました。そこで 64bit 版 Windows7 環境へのインストール手順をまとめた「インストールマニュアル」を別紙として用意しました。64bit 環境を利用している場合は上記 HP と合わせてご覧いただければうれしいです。

1.4. ネーミングサーバの起動

- ① まずはネーミングサーバを立ち上げます。画面左下のウィンドウズボタンをクリックし、「すべてのプログラム」「OpenRTM-aist 1.1 (64-bit)」「C++」「tools」と辿って「Start Naming Service」をクリックして起動しましょう。

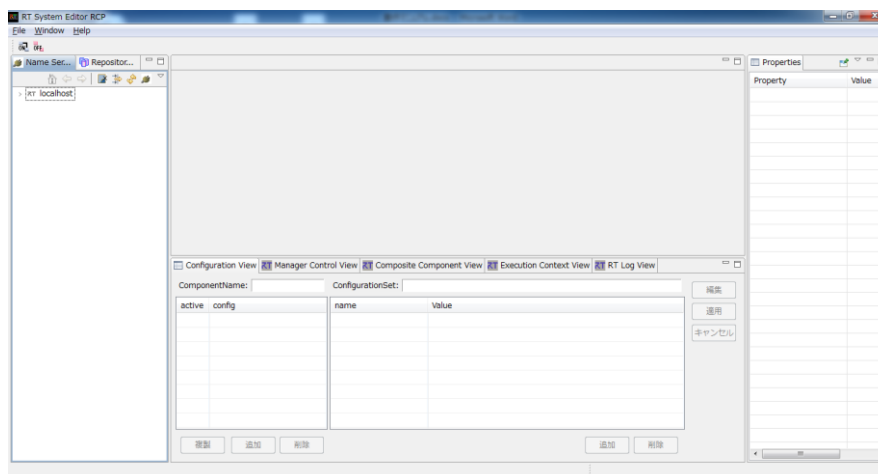


- ② 以下のようにネーミングサーバが立ち上がりますので、邪魔にならないように最小化しておきましょう。

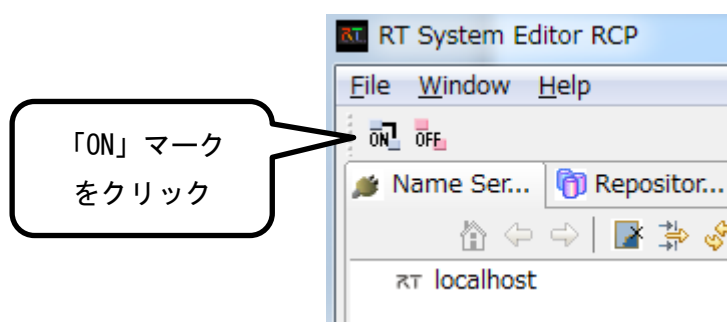


1.5. RT System Editor の起動

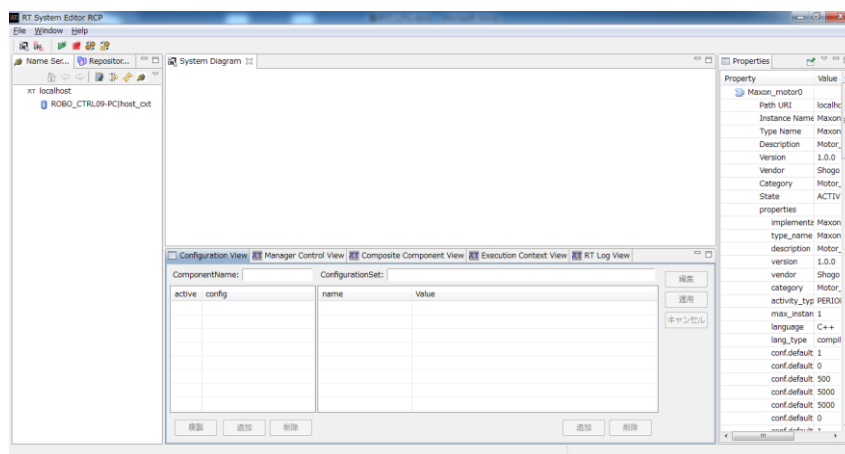
- ① 今度は RT System Editor を立ち上げます。画面左下のウィンドウズボタンをクリックし、「すべてのプログラム」「OpenRTM-aist 1.1(64bit)」「C++」「tools」と辿って「RTSystemEditor」をクリックして起動しましょう。



- ② ウィンドウ左上の「ON」マークをクリックして System Diagram を開きます。



- ③ System Diagram が立ち上がりました。



2. モータドライバ RTC



2.1. モータドライバ RTC の概要

まずはモータドライバ RTC の概要を以下に示します。このコンポーネントはマクソンジャパン（株）が公開しているライブラリとサンプルプログラムの一部を利用しています。

| RTC 名 | Maxon_motor | | |
|-------|--|------------|--|
| 構成 |  | | |
| | Data Port | | |
| | ポート種類 | ポート名称 | データ型 |
| | InPort | Target_Pos | TimedLong |
| | OutPort | Now_Pos | TimedLong |
| | Configuration Parameter（主要なもの） | | |
| | 名称 | 初期値 | 意味 |
| | abs_mode | 1 | 角度の指示方法を選択できます。 0：現在角度からの相対角度指示 1：EPOS 起動時を原点とした絶対角度指示 |
| | gear_para | 90 | 各モータのギア比に応じて調整するパラメータです。 |
| | range_limit | 90 | 異常動作を防ぐために、絶対値がこの値以上の角度指示は無視されます。 |
| | usNodeId | 1 | 複数の EPOS を利用する場合に設定する ID です。 |

2.2. 使用するハードウェア

この RTC で制御するハードウェアはマクソンジャパン（株）で販売されているモータドライバ EPOS2 24/2 です。接続するモータは同社の DC モータである A-max モータを使用し、MR エンコーダとスパーギアヘッドを取り付けています。使用したモータを構成する各要素の名称と型番は以下の通りです。

| | | | |
|-------------|------------|--------|---|
| DC モータ | 名称 | 型番 |  |
| | A-max16 | 110072 | |
| | GS16A-41:1 | 325066 | |
| | MR-256PPR | 228182 | |
| モータ ドライバ | EPOS 24/2 | |  |

追尾カメラとして利用する場合は・・・

EPOS 24/2 と PC は USB で接続します。EPOS 24/2 側は mini B 端子です。

追尾カメラモジュールではパン軸・チルト軸それぞれにこのモータとドライバを使用しているので、同様のモジュールを作る場合は上記の商品が 2 組必要なので注意が必要です。

2.3. 設定用ソフトウェアのダウンロード

買ったばかりのモータを動かす為には少し準備が必要です。まずは下記の URL が示すマクソンジャパン（株）の HP を表示しましょう。

<http://www.maxonjapan.co.jp/index.htm>

- ① 2012 年 11 月 15 日現在では以下の画面が表示されます。ページの構成が変化している場合は適宜読み替えてください。ページ上部のメニューから「製品」をクリックします。



- ② 下図のような画面になったら、左側の「maxon motor control (control electronics)」をクリックしてください。



- ③ 下図のような画面になったら「カタログデータ、各種マニュアルのダウンロードはこちらから」をクリックしてみましょう。



- ④ 現れたページの中央部「EPOS2」と書かれた周辺に以下のようなダウンロード表があります。以降はここから必要なプログラムをダウンロードしましょう。参照するのは「EPOS2 24/2」の列です。

| EPOS2マニュアル、ソフトウェア | | | EPOS2 24/1 | EPOS2 24/2 | EPOS2 24/5 | EPOS2 24/5 | EPOS2 Module 36/2 | EPOS2 50/5 | EPOS 70/10 | EPOS2 70/10 |
|--------------------------------------|---|--|------------|------------|------------|------------|-------------------|------------|------------|-------------|
| タイトル | 説明 | | | | | | | | | |
| ゲッティング・スタート | はじめての運転時の設定・接続 | | | | | | | | | |
| ハードウェア・リファレンス | ハードウェアの説明 | | | | | | | | | |
| ケーブル・スターティング・セット | コネクタと別売ケーブルの説明 | | | | | | | | | |
| Firmware Specification | ファームウェアの説明(英語) | | | | EPOS | | EPOS2 | | | |
| Communication Guide | 通信プロトコル(CAN, RS232)の説明(英語) | | | | EPOS | | EPOS2 | | | |
| EPOS Studio とファームウェア | 初期設定ソフト(886MB) | | | | | | | | | |
| 最新のファームウェア・ファイル | 上記EPOS Studioに含まれていますが、ここから個別にダウンロード可能です。 | | | | | | | | | |
| VisualC++, Windows DLL とサンプル、解説 | Microsoft Visual Studio Visual C++ 2005 and 6.0 | | | | | | | | | |
| VisualBasic, Windows DLL とサンプル、解説 | Microsoft Visual Studio Visual Basic 2005 and 6.0 | | | | | | | | | |
| BolandC++, Windows DLL とサンプル、解説 | C++ Builder 5.0 and Turbo C++10.0 | | | | | | | | | |
| Delphi, Windows DLL とサンプル、解説 | Delphi 10.0 and 4.0 | | | | | | | | | |
| LabView, Windows DLL とサンプル、解説 | LabVIEW 7.1 and 2010 | | | | | | | | | |
| LabWindows/CVI, Windows DLL とサンプル、解説 | LabWindows/CVI 7.1 and 2010 | | | | | | | | | |

- ⑤ 一覧から「ゲッティング・スタート」をダウンロードしてください。この pdf ファイルでは EPOS の初期設定方法について親切に説明されています。基本的には指示通りに操作すれば設定できるので、以降の初期設定はこのファイルを参照してください。次ページでは「ゲッティング・スタート」の補足説明をしてありますので合わせて参照することをお勧めします。

| EPOSマニュアル、ソフトウェア | | | | | | | | | |
|--------------------------------------|---|--------------|---------------|--------------|---------------|-------------------------|---------------|---------------|----------------|
| タイトル | 説明 | EPOS 24/1 | EPOS2 24/2 | EPOS 24/5 | EPOS2 24/5 | EPOS2 Module 36/2 | EPOS2 50/5 | EPOS 70/10 | EPOS2 70/10 |
| ゲッティング・スタート | はじめての運転時の設定・接続 | | | | | | | | |
| ハードウェア・リファレンス | ハードウェアの説明 | | | | | | | | |
| ケーブル・スターティング・セット | コネクタと別売ケーブルの説明 | | | | | | | | |
| Firmware Specification | ファームウェアの説明(英語) | | | | | | | | |
| Communication Guide | 通信プロトコル(CAN, RS232)の説明(英語) | | | | | | | | |
| EPOS Studio とファームウェア | 初期設定ソフト(886MB) | | | | | | | | |
| 最新のファームウェア・ファイル | 上記EPOS Studioに含まれていますが、ここから個別にダウンロード可能です。 | | | | | | | | |
| VisualC++, Windows DLL とサンプル、解説 | Microsoft Visual Studio Visual C++ 2005 and 6.0 | | | | | | | | |
| VisualBasic, Windows DLL とサンプル、解説 | Microsoft Visual Studio Visual Basic 2005 and 6.0 | | | | | | | | |
| BolandC++, Windows DLL とサンプル、解説 | C++ Builder 5.0 and Turbo C++10.0 | | | | | | | | |
| Delphi, Windows DLL とサンプル、解説 | Delphi 10.0 and 4.0 | | | | | | | | |
| LabView, Windows DLL とサンプル、解説 | LabVIEW 7.1 and 2010 | | | | | | | | |
| LabWindows/CVI, Windows DLL とサンプル、解説 | LabWindows/CVI 7.1 and 2010 | | | | | | | | |

「ゲッティング・スタート」
をクリック

2.4. EPOS2 24/2 の配線と初期設定

前述したように配線や初期設定に関しては「ゲッティング・スタート」の解説が詳しいのでそちらを参照してください。ゲッティング・スタートで指示されている手順がすべて完了したら、このマニュアルを再び参照して以降の「2.5. RTC を利用した動作確認」に進みましょう。以下ではゲッティング・スタートの概略と補足を述べます。

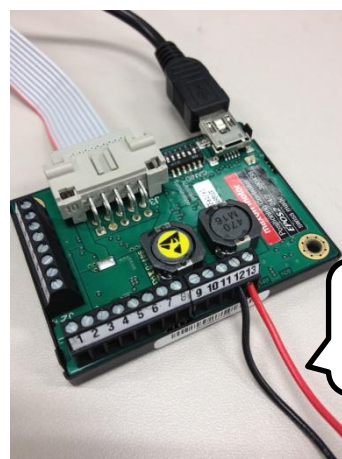
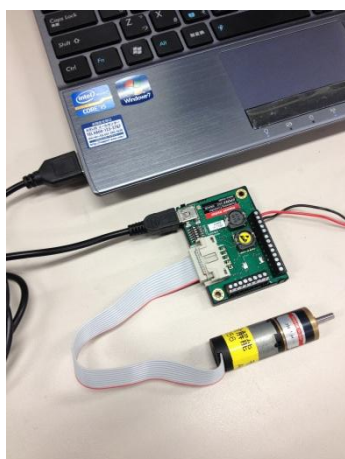
補足説明を加えた大まかな流れとしては

- ① 商品に付属の DVD-ROM を用いて PC にソフトウェアをインストールします。

途中でインストールの種類を聞かれますが「Typical」で構わないでしょう。

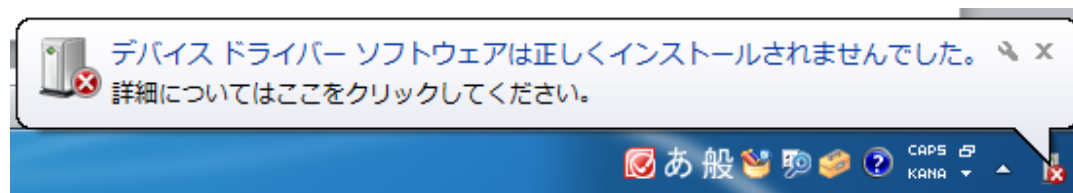
- ② 配線を行います。

配線らしき配線は電源となる DC+12V と GND の 2 線をつけるだけです。念のため、以下に配線した EPOS の外観を示します。電源には 12V 1.5A の AC アダプタを利用しました。



- ③ インストールした EPOS Studio でシステム設定をします。

EPOS の電源を入れても、そのままでは PC 上でドライバが正常に認識されないという問題が起きました。これに対しては次ページ以降の手順で解決できます。



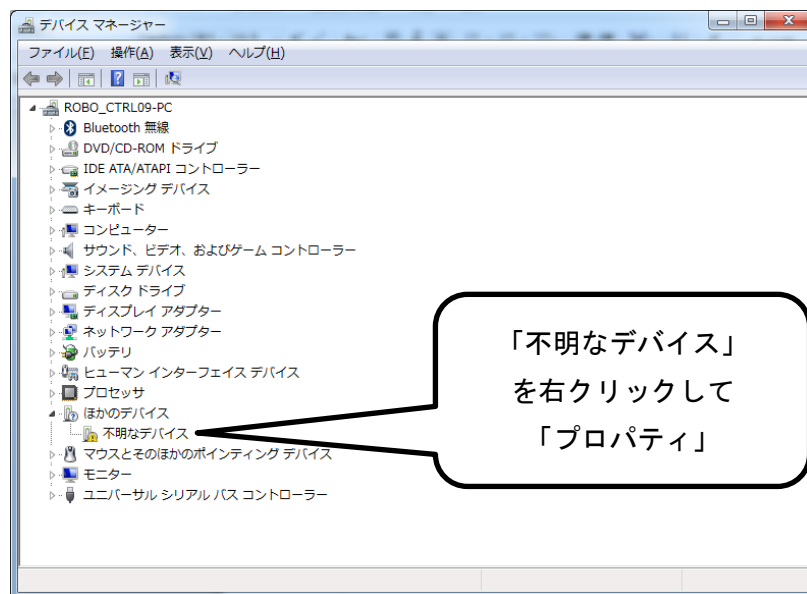
- ④ 最後に、モータの制御ゲイン調整を行います。

使用しているモータの使用値を参考にして指示通りに行えば問題ありません

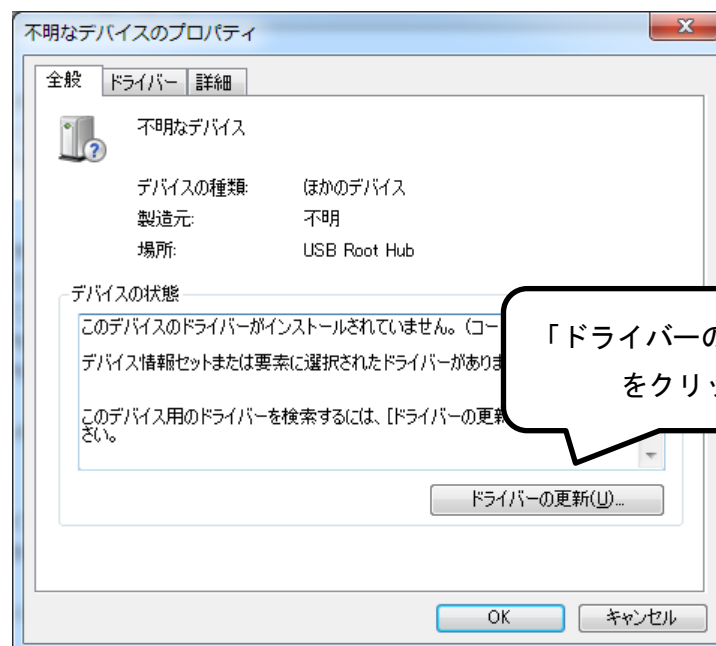
デバイスドライバの設定方法

i) 「コントロールパネル」「ハードウェアとサウンド」「デバイスマネージャ」という順に進んでデバイスマネージャを起動させます。

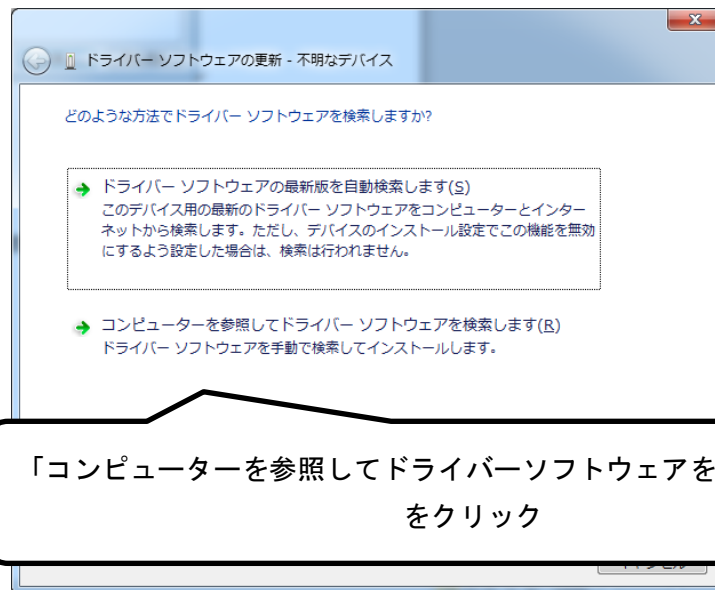
ii) 一覧のうち「ほかのデバイス」下に「不明なデバイス」がありますので右クリックして「プロパティ」を選択します。



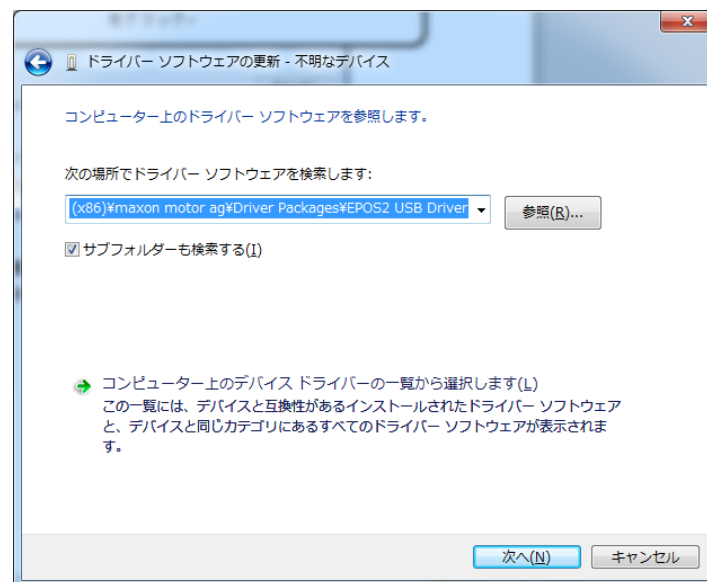
iii) 「ドライバーの更新(U)」をクリックしてください。



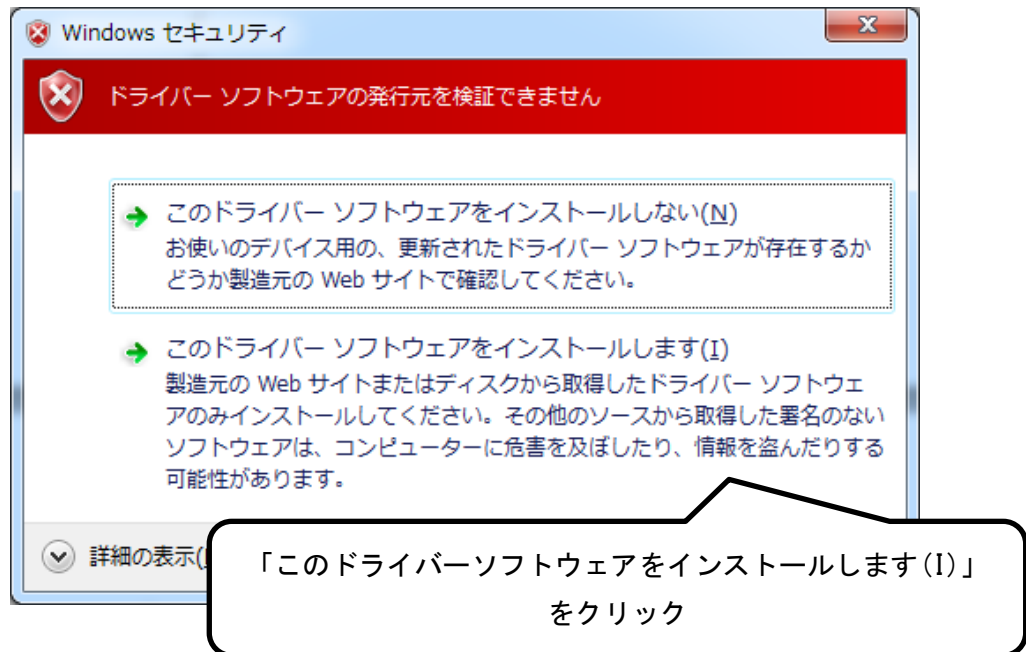
iv) 「コンピューターを参照してドライバーソフトウェアを検索します (R)」をクリックしてください。



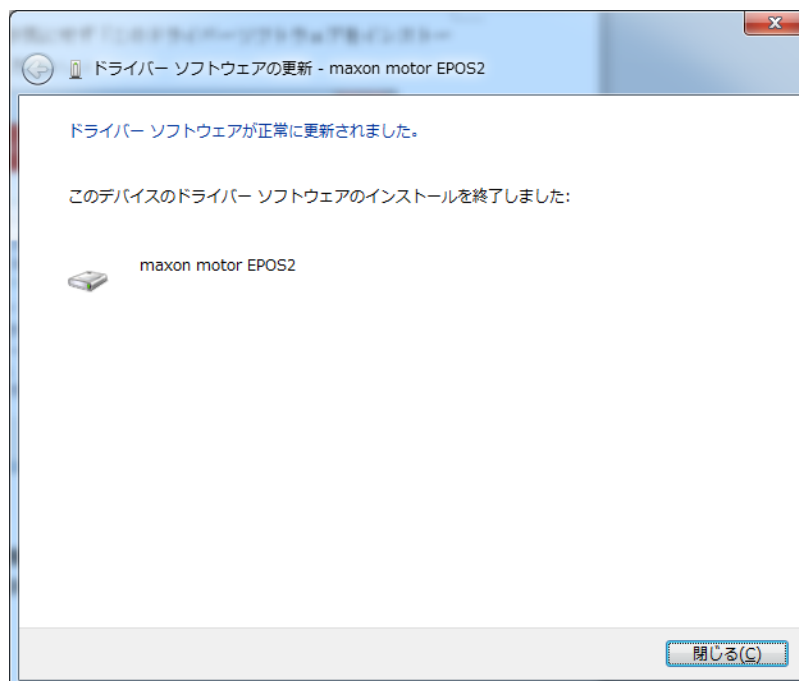
v) 参照をクリックしてドライバの場所を指定します。デフォルトの場所にインストールしているならば以下のパスとなります。指定したら「次へ」をクリックしてください。
「C:\Program Files (x86)\maxon motor ag\Driver Packages\EPOS2 USB Driver」



vi) 以下のような警告が出ますが気にせず「このドライバーソフトウェアをインストールします(I)」をクリックしてください。



vii) 以下の画面が表示されればデバイスドライバが正常にインストールされました。「閉じる(C)」をクリックしたら、本来の手順に戻って設定を進めましょう。



2.5. RTC を利用した動作確認

それでは、いよいよ RTC を利用した動作を確認してみましょう。ネーミングサービスと RT System Editor はあらかじめ起動しておいてください。（操作は 1.4. ～1.5 にて解説済）

- ① 以下の URL で示すダウンロードページから「モータドライバ RTC」をダウンロードして任意のフォルダに保存してください。先頭のフォルダ名は「Maxon_motor」です。

<http://www.meo.shibaura-it.ac.jp/matsuhira/RTM.html>

| | | | |
|-----------------------|------------------|-------------|------|
| .metadata | 2012/11/13 16:36 | ファイル フォル... | |
| Maxon_motor | 2012/11/14 1:42 | ファイル フォル... | |
| Tracking_Controller | 2012/11/14 1:42 | ファイル フォル... | |
| URG_scan | 2012/11/14 3:15 | ファイル フォル... | |
| systemeditor0.log | 2012/11/15 20:33 | テキスト ドキュ... | 2 KB |
| systemeditor0.log.lck | 2012/11/15 20:31 | LCK ファイル | 0 KB |

- ② このコンポーネントに必要なマクソンジャパン(株)のライブラリファイルを以下の URL からダウンロードしてください。いくつかの製品に対するダウンロードの一覧が表示されますが、参照するのはページ中央部の「EPOS2」の項目で「EPOS224/2」と書かれた列の中から「Visual C++, Windows DLL とサンプル、解説」の行にあるファイルです。

http://www.maxonjapan.co.jp/product_mmc.htm

| EPOSマニュアル、ソフトウェア | | EPOS 24/1 | EPOS2 24/2 | EPOS 24/5 | EPOS2 24/5 | EPOS2 Module 36/2 | EPOS2 50/5 | EPOS 70/10 | EPOS2 70/10 |
|---------------------------------|--|-----------|------------|-----------|------------|-------------------|------------|------------|-------------|
| タイトル | 説明 | | | | | | | | |
| ゲッティング・スタート | はじめての運転時の設定・接続 | | | | | | | | |
| ハードウェア・リファレンス | ハードウェアの説明 | | | | | | | | |
| ケーブル・スターティング・セット | コネクタと別売ケーブルの説明 | | | | | | | | |
| Firmware Specification | ファームウェアの説明(英語) | | | | | | | | |
| Communication Guide | 通信プロトコル(CAN, RS232)の説明(英語) | | | | | | | | |
| EPOS Studio とファームウェア | 初期設定ソフト(886MB) | | | | | | | | |
| 最新のファームウェア・ファイル | 上記 EPOS Studio に含まれています ここから個別にダウンロード可能です。 | | | | | | | | |
| VisualC++, Windows DLL とサンプル、解説 | Microsoft Visual Studio Visual C++ 2005 and 6.0 | | | | | | | | |
| VisualBasic, Windows DLL とサ | Microsoft Visual Studio | | | | | | | | |

ここをクリックして
「保存」

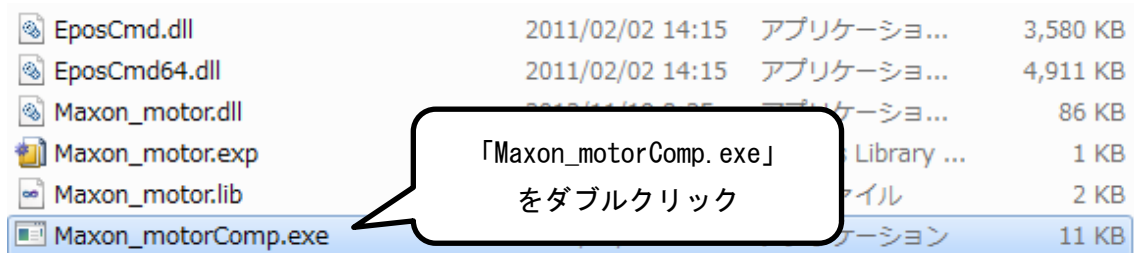
- ③ 保存したファイルを解凍すると「Microsoft Visual C++」というフォルダが生成されるのでフォルダ内の「Example VC++ 2005」を開いてください。以下に示すファイルがあります。今回利用するのはこの中の「Definitions.h」「EpoCmd.lib」「EposCmd64.lib」「EposCmd.dll」「EposCmd64.dll」のみであり、全部で5つとなります。

| | | |
|--------------------|------------------|---------------------|
| res | 2012/11/20 10:50 | ファイル フォル... |
| Definitions.h | 2011/02/02 14:15 | C/C++ Header |
| Demo_WinDLL.cpp | 2010/12/13 13:11 | C++ Source |
| Demo_WinDLL.exe | 2011/01/12 10:22 | アプリケーション |
| Demo_WinDLL.h | 2010/12/13 13:11 | C/C++ Header |
| Demo_WinDLL.rc | 2010/12/13 13:11 | Resource Script |
| Demo_WinDLL.sln | 2011/01/12 10:22 | Microsoft Visual... |
| Demo_WinDLL.vcproj | 2011/01/12 10:22 | VC++ Project |
| Demo_WinDLLDlg.cpp | 2011/01/12 10:22 | C++ Source |
| Demo_WinDLLDlg.h | 2011/01/12 10:22 | C/C++ Header |
| EposCmd.dll | 2011/02/02 14:15 | アプリケーショ... |
| EposCmd.lib | 2011/02/02 14:15 | LIB ファイル |
| EposCmd64.dll | 2011/02/02 14:15 | アプリケーショ... |
| EposCmd64.lib | 2011/02/02 14:15 | LIB ファイル |
| resource.h | 2010/12/13 13:11 | C/C++ Header |
| StdAfx.cpp | 2010/12/13 13:11 | C++ Source |
| StdAfx.h | 2010/12/13 13:11 | C/C++ Header |
| vxlapl.dll | 2010/12/22 15:27 | アプリケーショ... |
| vxlapl64.dll | 2010/12/22 15:27 | アプリケーショ... |

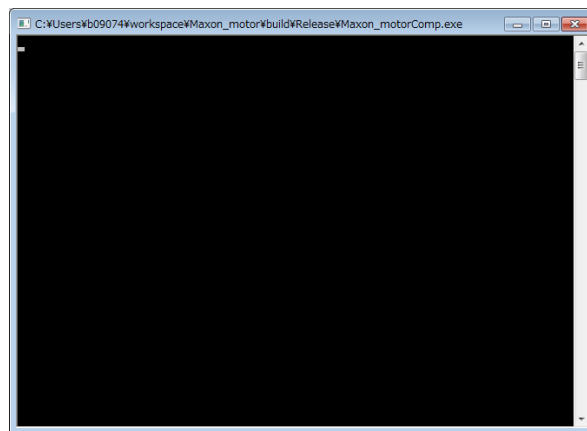
- ④ 以下の表にしたがって該当ファイルを①で開いた「Maxon_motor」フォルダ内の指定の場所にコピーしてください。

| ファイル名 | コピーする場所 |
|---------------|--|
| Definitions.h | Maxon_motor¥build |
| EpoCmd.lib | Maxon_motor¥build |
| EposCmd64.lib | Maxon_motor¥build |
| EposCmd.dll | Maxon_motor¥build¥Release (あるいは Debug) |
| EposCmd64.dll | Maxon_motor¥build¥Release (あるいは Debug) |

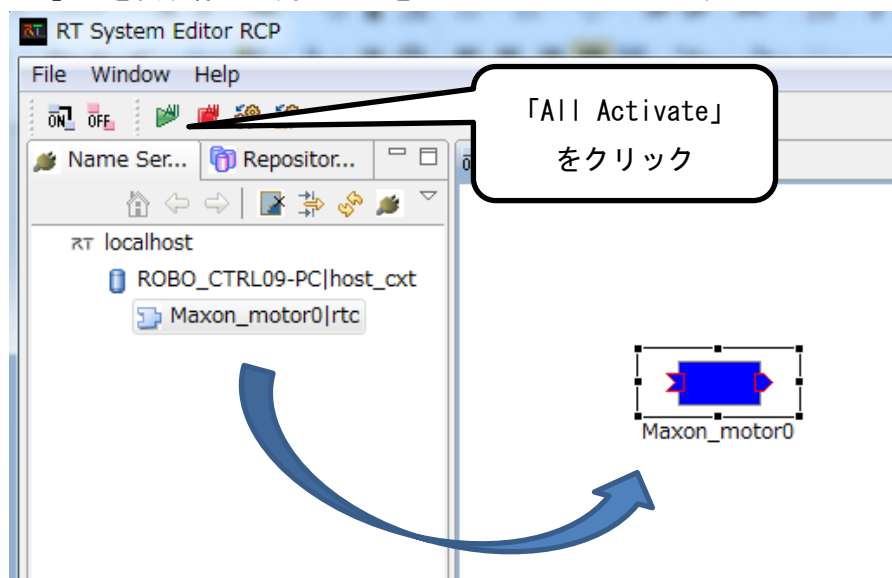
- ⑤ 「Maxon_motor」フォルダ内を「build」「Release」と辿ると「Maxon_motorComp.exe」という実行ファイルがあるのでダブルクリックして実行してください。



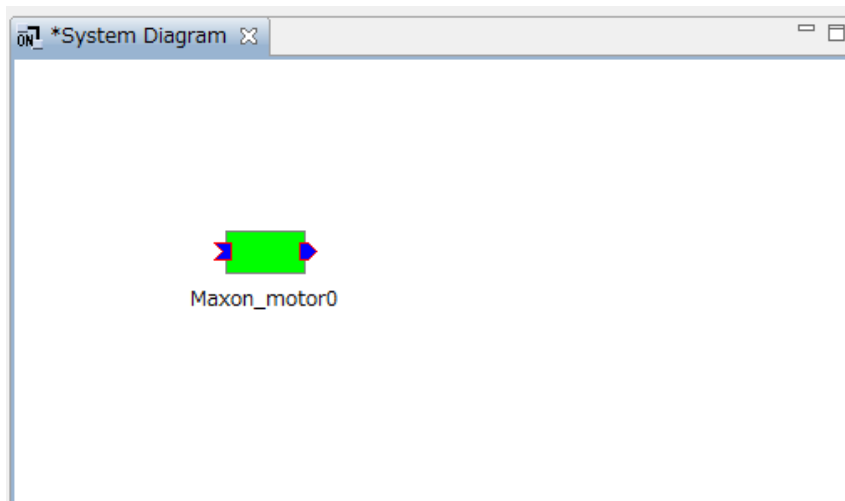
- ⑥ 真っ黒なウィンドウが立ち上がりますので、そのままにしておきましょう。



- ⑦ Maxon_motorComp.exe でウィンドウを立ち上げると RT System Editor 上に起動したコンポーネントが表示されるので System Diagram 上にドラック&ドロップして「All Activate」のを表す緑の三角マークをクリックしてみましょう。



- ⑧ System Diagram 上で RTC の表示が緑色に変化すれば接続成功です。この状態で degree 単位の角度指示値を与えると、モータは EPOS 起動時の位置を原点として指定の角度だけ動作します。エラーが出た場合は先ほど真っ黒だったコンソール上にエラーメッセージが現れますので参考にしてみてください。

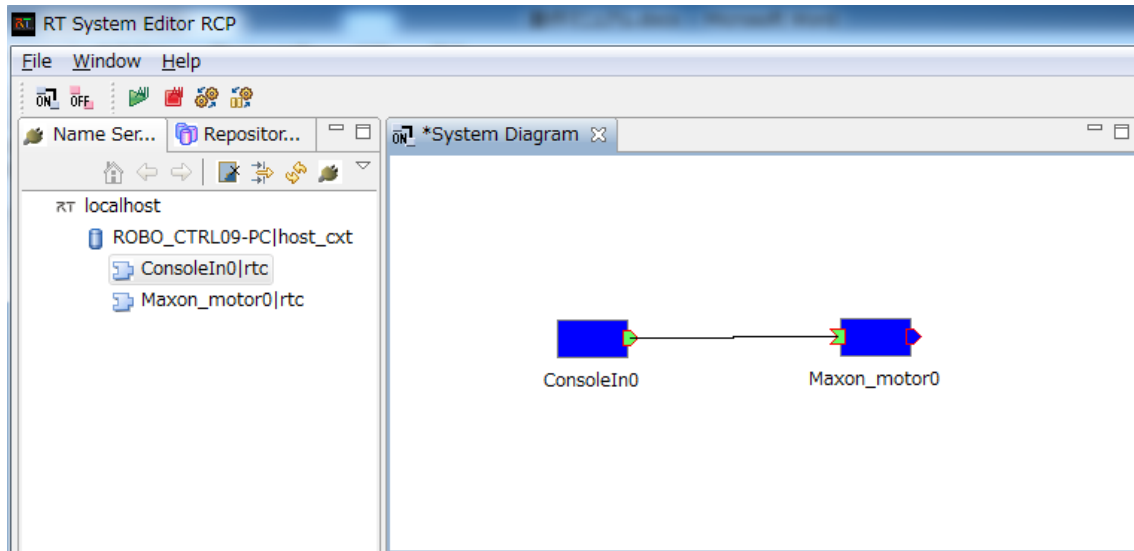


- ⑨ モータドライバ RTC の InPort は TimedLong 型のデータポートですので OpenRTM-aist-1.1.0 に付属のサンプルである「ConsoleIn」を利用して角度指示が可能です。実際に画面左下のウィンドウズボタンをクリックし、「すべてのプログラム」「OpenRTM-aist 1.1(64bit)」「C++」「components」「examples」と辿って「ConsoleInComp.exe」をクリックして起動し、試してみましょう。

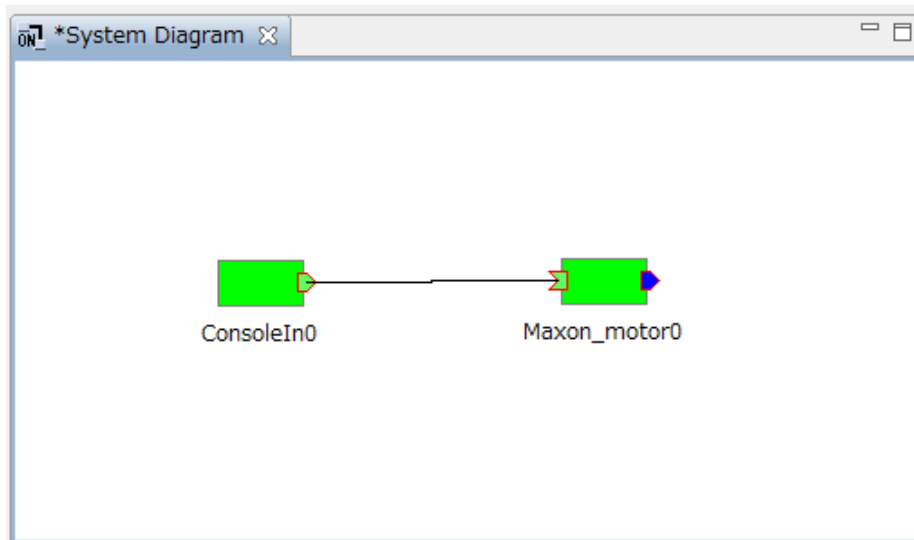
```

ConsoleInComp.exe
corba.id: omniORB
exec_cxt.periodic.type: PeriodicExecutionContext
exec_cxt.periodic.rate: 1000
exec_cxt.evdriven.type: EventDrivenExecutionContext
logger.enable: NO
logger.log_level: INFO
naming.enable: YES
naming.type: corba
naming.formats: %h.host_cxt/%n.rtc
naming.names: ROBO_CTRL09-PC.host_cxt/ConsoleIn0.rtc
config_file:
port.outport.dataport:
port.outport.out:
=====
Port0 (name): ConsoleIn0.out
=====
- properties -
port.port_type: DataOutPort
dataport.data_type: IDL:RTC/TimedLong:1.0
dataport.subscription_type: flush,new,periodic
dataport.dataflow_type: push,pull
dataport.interface_type: corba_cdr
=====
    
```

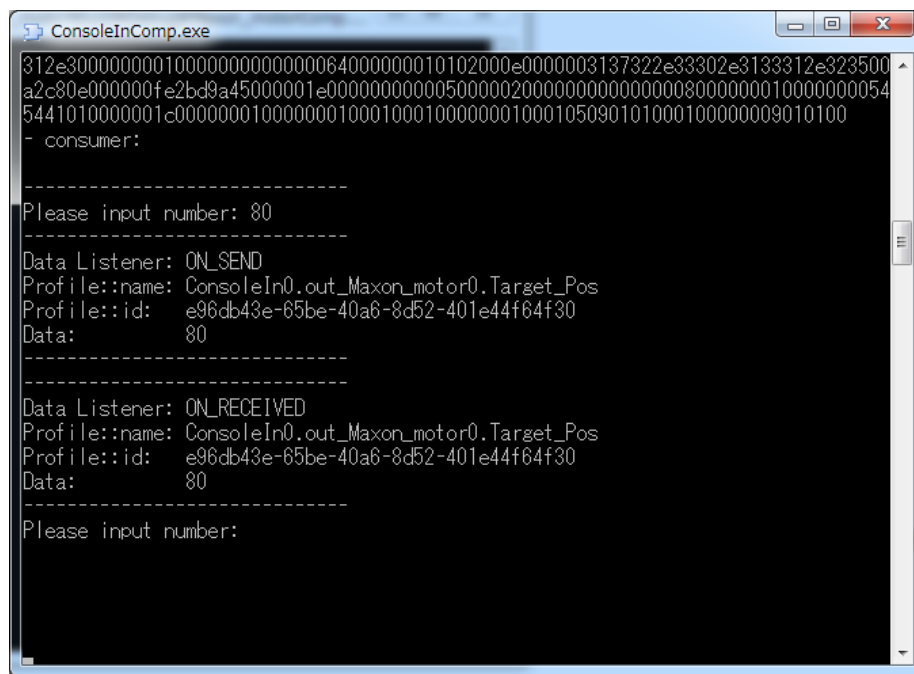
- ⑩ 「ConsoleInComp.exe」を System Diagram 上にドラッグ&ドロップし、先ほど起動した「Maxon_motor」の InPort と接続してみましょう。接続は ConsoleIn の OutPort から Maxon_motor の InPort へドラッグすることで繋がります。



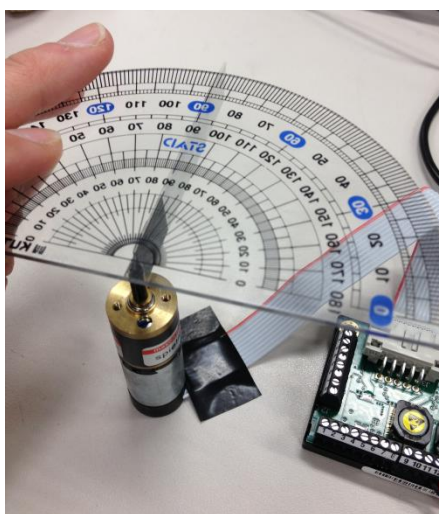
- ⑪ 再び「All Activate」を示す緑の三角マークをクリックしてみましょう。以下のように両方のコンポーネントが緑色に変化すれば起動は成功です。



- ⑫ テストとして ConsoleIn のウィンドウ上で「80」と入力し Enter を押しましょう。Enter を押すと同時にモータが動作するはずですが、初期設定では 90° 以上の値は受け付けられないようになっていますが、コンフィギュレーションパラメータを変化させることでこの制限も変更することができます。



- ⑬ モータが動作したら、次はパラメータの調整を行います。先ほど「80」と入力したにも関わらず、あまり回転しない、あるいは回転しすぎるといった場合はコンフィギュレーションパラメータの「gear_para」を調節することで解決できます。
- 簡単な確認方法として、モータ先端に厚紙等を張り付け、分度器を当てて「80」と入力しましょう。もし到達した位置が 80° 以上であった場合は gear_para の値を小さく、 80° 以下であった場合は gear_para の値を大きくすることで調整してください。



- ⑭ モータの動作速度を変化させたい場合は以下のコンフィギュレーションパラメータで変更可能です。なお変更した結果デバイスの接続時に適応されますので、速度や変えるには一度 Deactivate してから再接続してください。

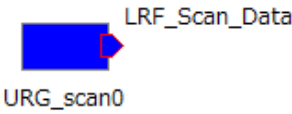
| | |
|-----------------------|-----|
| ulProfileAcceleration | 加速度 |
| ulProfileDeceleration | 減速度 |
| ulProfileVelocity | 速度 |

以上でモータドライバ RTC の解説は終了です。思い通りにモータを回転させることはできたでしょうか。

3. URG 計測 RTC



3.1. URG 計測 RTC の概要

まずは URG 計測 RTC の概要を以下に示します。このコンポーネントは北陽電機（株）が公開しているライブラリとサンプルプログラムの一部を利用しています。

| RTC 名 | URG_scan | | | |
|-------|--|---------------|-------------------------|-------------------|
| 構成 |  | | | |
| | Data Port | | | |
| | ポート種類 | ポート名称 | データ型 | 意味 |
| | OutPort | LRF_Scan_Data | RangeData | URG の距離データを送信します。 |
| | Configuration Parameter | | | |
| | 名称 | 初期値 | 意味 | |
| | comNo | com1 | URG と接続する COM 番号を設定します。 | |

3.2. 使用するハードウェア

この RTC で制御するハードウェアは北陽電機（株）で販売されている測域センサである URG シリーズです。動作を確認できたのは「URG-04LX-UG01」と「URG-04LX」の 2 種類となっています。どちらか好きな方のセンサを利用してください。

| | | | |
|-------------------|------|---------------------------|--|
| URG-04LX -UG01 | 測距範囲 | 20～5, 600mm |  |
| | | 240° | |
| | 測距精度 | 60～1, 000mm ±30mm | |
| | | 1, 000～4, 095mm 距離の±3% | |
| URG-04LX | 測距範囲 | 60～4095mm |  |
| | | 240° | |
| | 測距精度 | 60～1000mm ±10mm | |
| | | 1000～4095mm 距離の±1% | |

どちらの測域センサにすればいいか

とりあえず追尾カメラを動作させたいなら URG-04LX-UG01（画像：上）は相対的に測距精度が劣るものの、追尾カメラを動作させる上では問題ない範囲です。電源も USB からのバスパワーのみで動作するので手軽に利用できる点がお勧めです。

追尾カメラ以外にも精度を必要とする用途で使用する場合は URG-04LX（画像：下）が適しているでしょう。ただし、DC5V を別に用意する必要があります。

3.3. デバイスドライバのダウンロード

URG を PC に接続する際のデバイスドライバをダウンロードします。まずは下記の URL が示す北陽電機（株）の HP を表示しましょう。

<http://www.hokuyo-aut.co.jp/02sensor/07scanner/download/index.html>

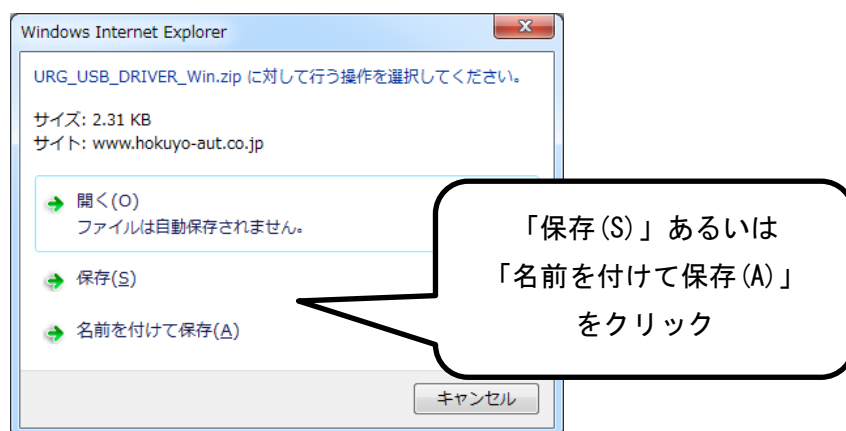
- ① 2012 年 11 月 15 日現在では以下の画面が表示されます。ページの構成が変化している場合は適宜読み替えてください。ページ中央の「商品ラインナップ」から使用する URG の写真をクリックしてください。



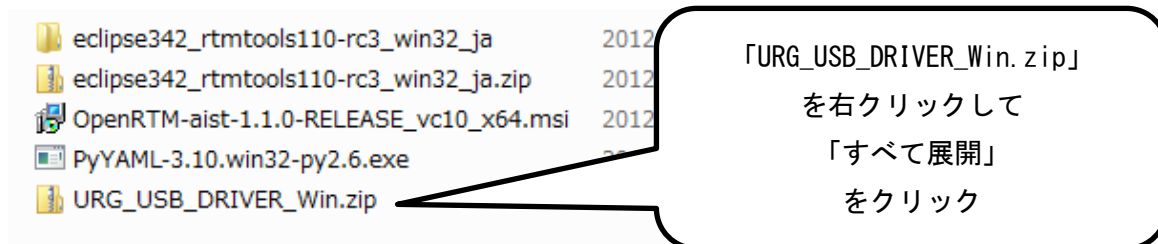
- ② 「ダウンロード」という項目に「URG-04LX-UG01 ドライバ: Windows 用 (32bit/64bit)」あるいは「URG-04LX ドライバ: Windows 用 (32bit/64bit)」があるのでクリックしてください。どちらか片方をインストールすれば前述した 2 種類の URG がどちらも正常に動作するので、ダウンロードするファイルの内容は同じようです。



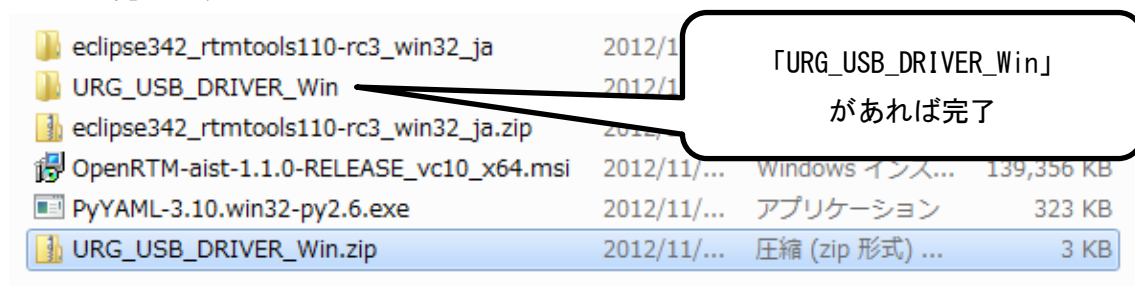
- ③ 以下のようなウィンドウが現れたら「保存(S)」あるいは「名前を付けて保存(A)」をクリックしてください。解説では「保存(S)」をクリックしました。



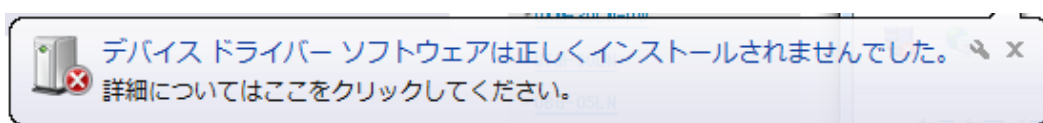
- ④ ダウンロードしたディレクトリを確認すると「URG_USB_DRIVER_Win.zip」というファイルがあるので右クリックして「すべて展開」をクリックしたら同じ場所に展開してください。



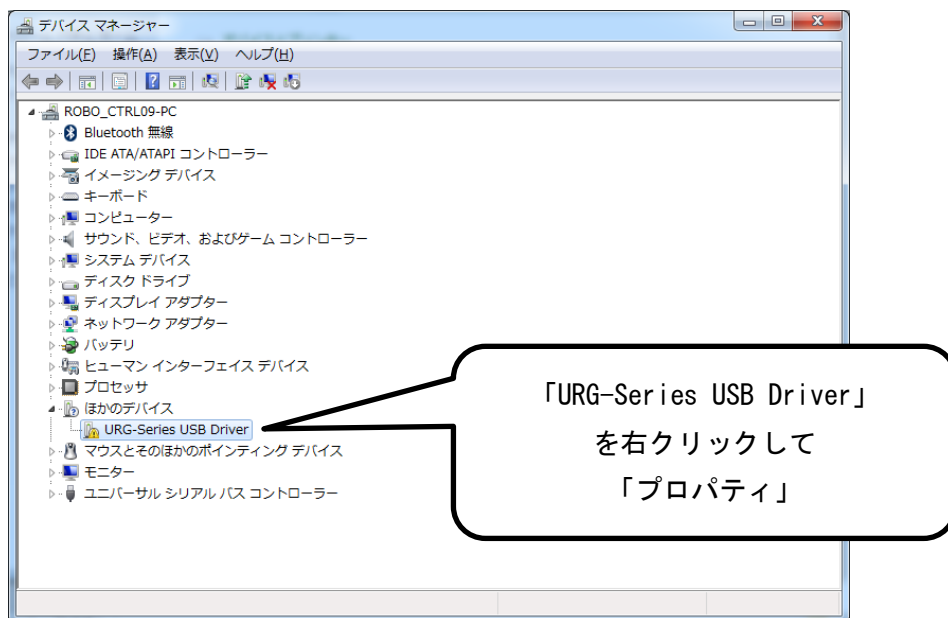
- ⑤ 展開したファイル「URG_USB_DRIVER_Win」が現れました。これでドライバのダウンロードは完了です。



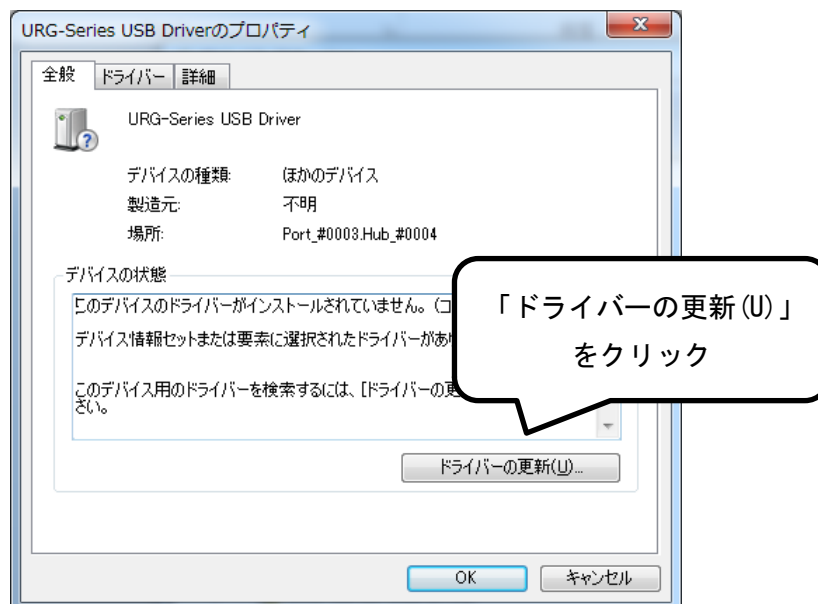
- ⑥ URG と PC を USB で接続して、ダウンロードしたドライバが正常に動作するか確認してみましょう。実際に URG と PC を USB で接続してください。最初に接続するときは以下のような表示が出てしまいます。



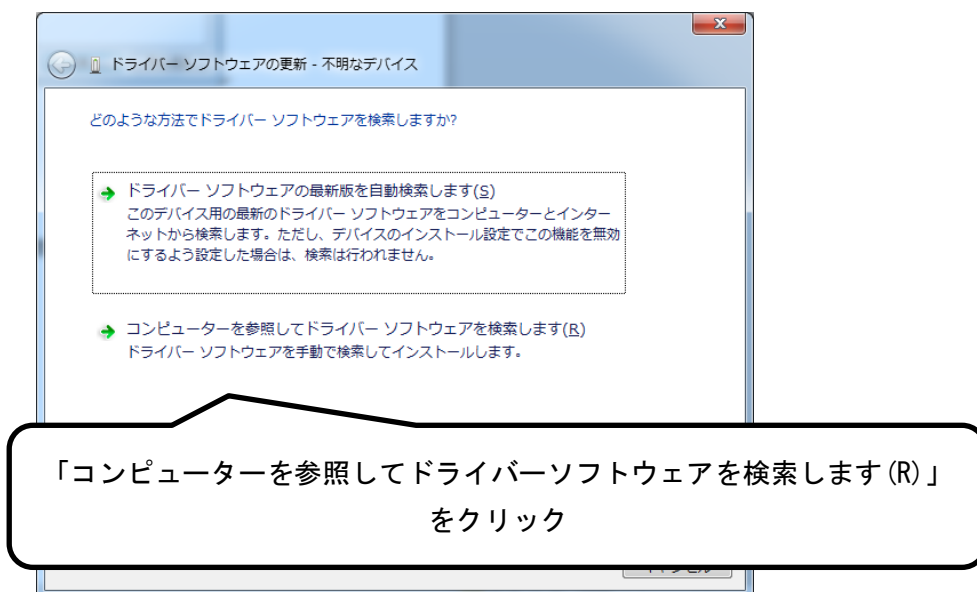
- ⑦ 正しくドライバを認識させるために、「コントロールパネル」「ハードウェアとサウンド」「デバイスマネージャ」という順に辿ってデバイスマネージャを起動させます。
- ⑧ 一覧のうち「ほかのデバイス」下に「URG-Series USB Driver」がありますので右クリックして「プロパティ」を選択します。



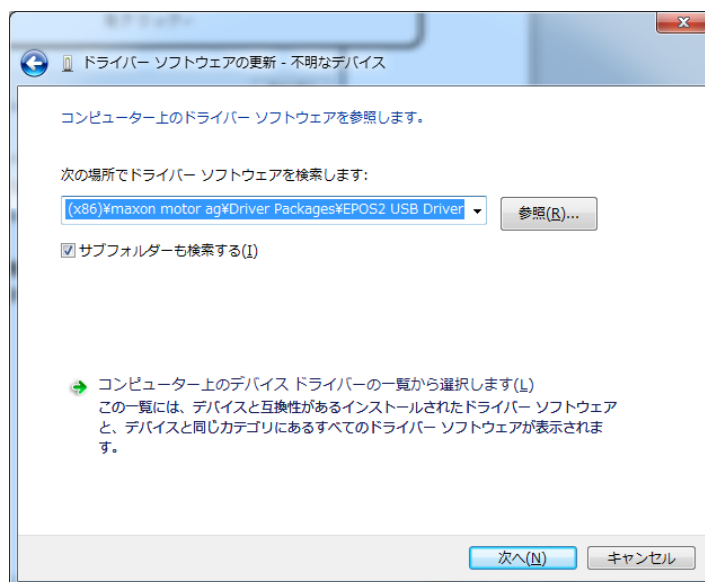
- ⑨ 「ドライバーの更新 (U)」をクリックしてください。



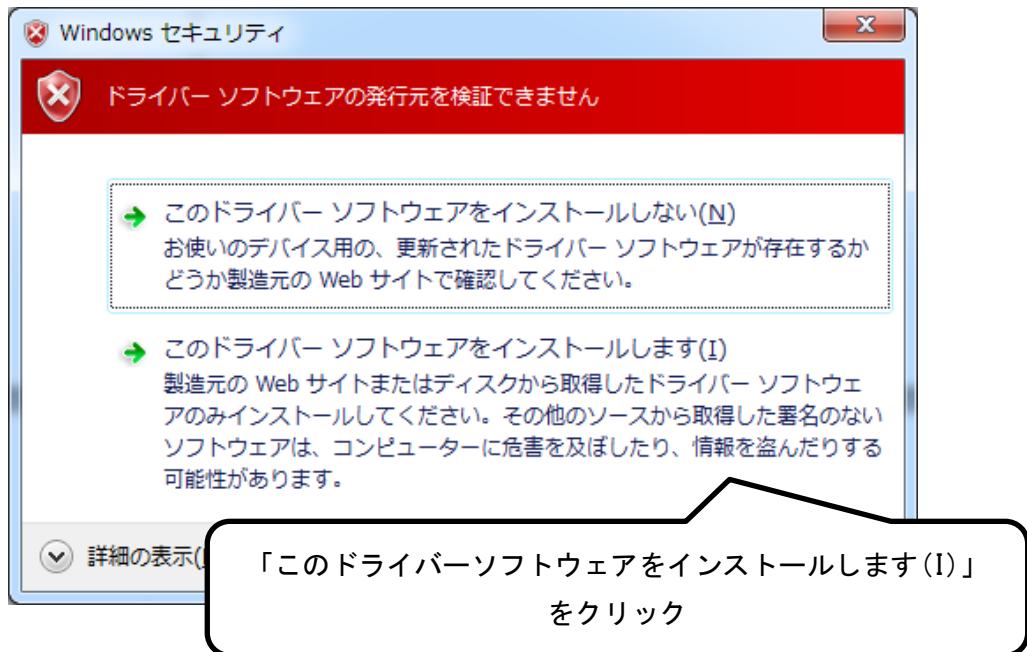
- ⑩ 「コンピューターを参照してドライバーソフトウェアを検索します(R)」をクリックしてください。



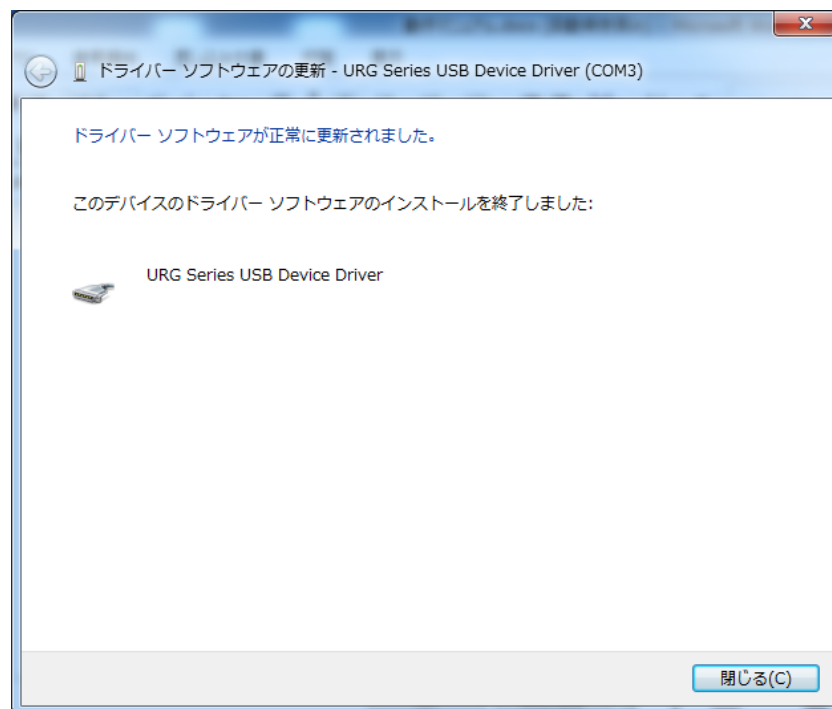
- ⑪ 「参照(R)」をクリックしてドライバの場所を指定します。先ほどドライバをダウンロードしたディレクトリ上から「URG_USB_DRIVER_Win」内の「URG_USB_Driver」を指定してください。指定したら「次へ」をクリックしてください。



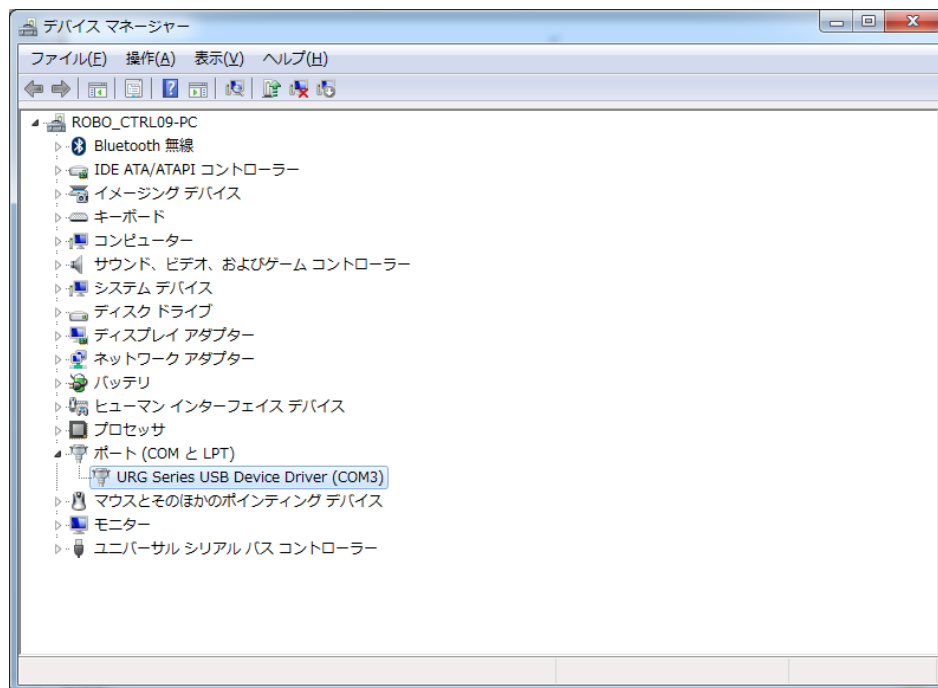
- ⑫ 以下のような警告が出ますが気にせず「このドライバーソフトウェアをインストールします(I)」をクリックしてください。



- ⑬ 以下の画面が表示されればデバイスドライバが正常にインストールされました。「閉じる(C)」をクリックしてください。








- ⑭ 再びデバイスマネージャを確認すると以下のように表示されます。括弧内にある COM ポート番号（ここでは COM3）をメモしておいてください。COM ポート番号はどの USB ポートを使用しているかによって異なります。







3.4. RTC ミドルウェアを利用した動作確認

それでは、いよいよ RTC を利用した動作を確認してみましょう。ネーミングサービスと RT System Editor はあらかじめ起動しておいてください。（操作は 1.4. ～1.5 にて解説済）

- ① プロジェクトページから URG 計測 RTC をダウンロードして任意のフォルダに保存してください。先頭のフォルダ名は「URG_scan」です。

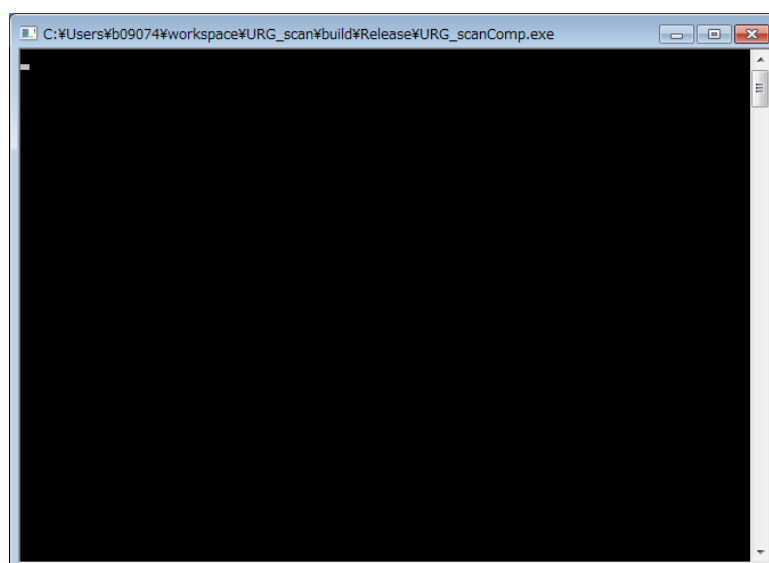
| | | | | |
|---|---------------------|------------------|-------------|------|
|  | .metadata | 2012/11/13 16:36 | ファイル フォル... | |
|  | Maxon_motor | 2012/11/14 1:42 | ファイル フォル... | |
|  | Tracking_Controller | 2012/11/14 1:42 | ファイル フォル... | |
|  | URG_scan | 2012/11/14 3:15 | ファイル フォル... | |
|  | systemeditor0.log | 2012/11/15 22:57 | テキスト ドキュ... | 7 KB |

- ② 保存したフォルダ内を「build」「Release」と辿ると「URG_scanComp.exe」という実行ファイルがあるのでダブルクリックして実行してください。

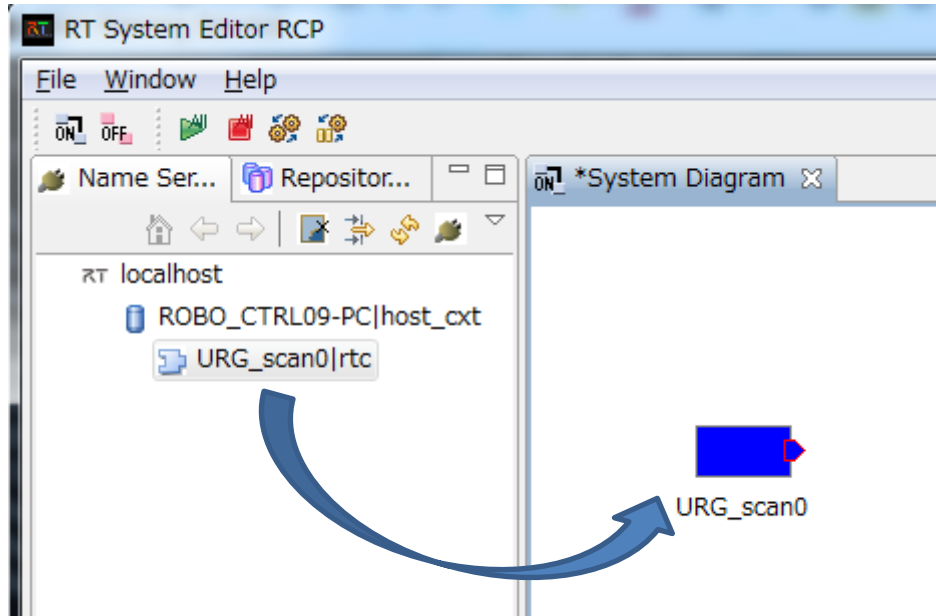
| | | | |
|---|------------------|-------------------|-------|
|  | URG_scan.dll | アプリケーション... | 63 KB |
|  | URG_scan.exp | ports Library ... | 1 KB |
|  | URG_scan.lib | 8 ファイル | 2 KB |
|  | URG_scanComp.exe | アプリケーション | 11 KB |

「URG_scanComp.exe」
をダブルクリック

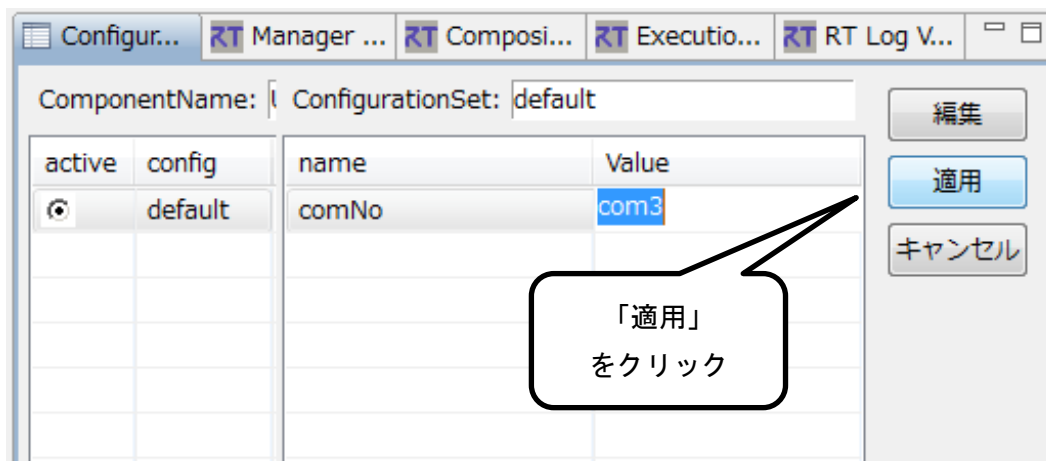
- ③ 真っ黒なウィンドウが立ち上がりますので、そのままにしておきましょう。



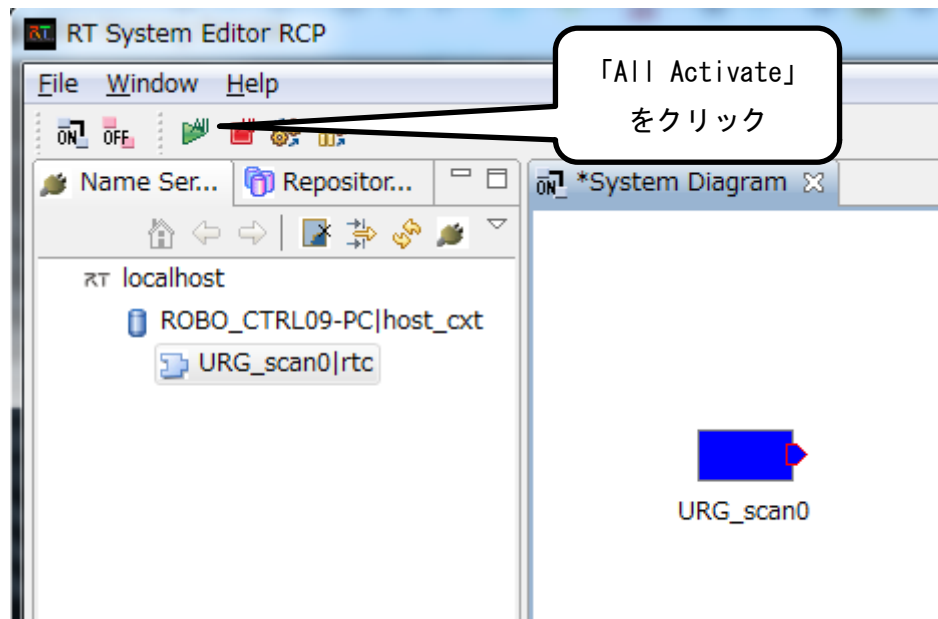
- ④ URG_scanComp.exe でウィンドウを立ち上げると RT System Editor 上に起動したコンポーネントが表示されるので System Diagram 上にドラック＆ドロップしましょう。



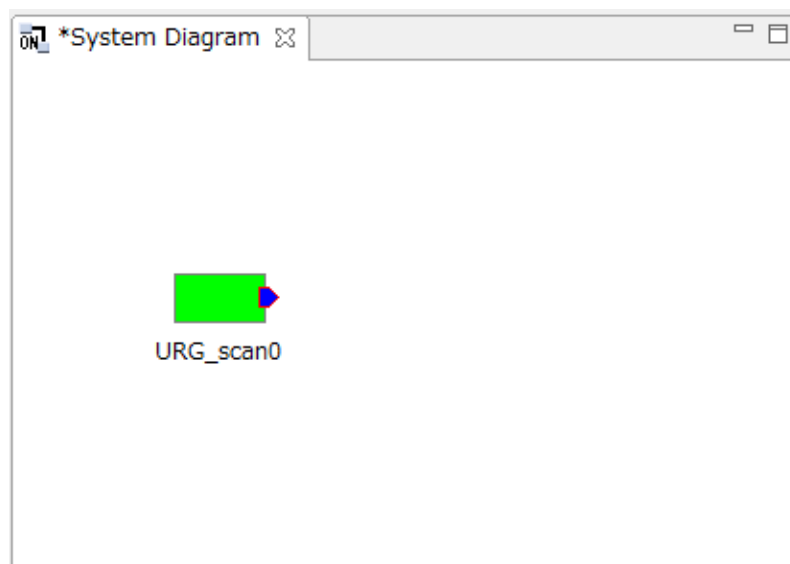
- ⑤ Activate する前に, System Diagram 上で「URG_scan」を選択し, RT System Editor 下部に表示されているコンフィギュレーションパラメータ上で COM 番号を設定しましょう。デフォルトでは「COM1」ですが, 先ほどデバイスマネージャ上で確認してメモを取った「COM3」に変更します。変更したら忘れずに「適用」をクリックしてください。



- ⑥ 「All Activate」のを表す緑の三角マークをクリックしてみましょう。



- ⑦ System Diagram上でRTCの表示が緑色に変化すれば接続成功です。赤色に変化した場合はエラーを意味します。先ほど真っ黒だったコンソール上にエラーメッセージが現れますので参考にしてみてください。



補足として

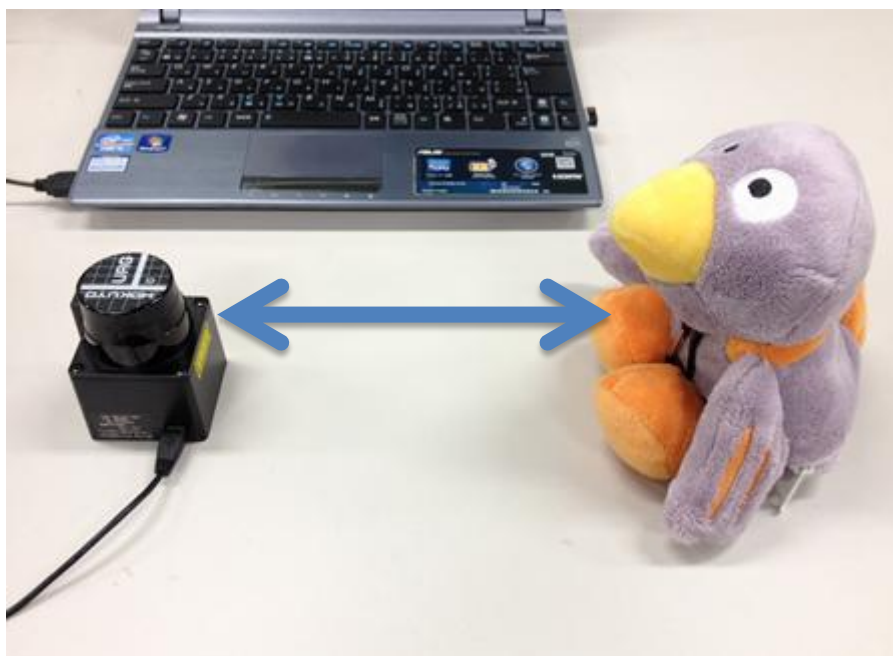
エラーが発生した場合、原因として最も怪しいのはCOMポート番号です。

再びデバイスマネージャで確認し、URG_scanのコンフィギュレーションパラメータが一致しているか確かめてください。

- ⑧ URG 計測 RTC が起動するとコンソール上に以下のような距離データが出力されます。表示される数値の単位は mm です。この表示は動作確認用に URG 前方の距離のみを出力しています。URG の前に何か物体を置いて移動させ、出力データが妥当な値であるか確認しましょう。

```

C:\Users\b09074\workspace\URG_scan\build\Release\URG_scanComp.exe
Front Data =303
Front Data =304
Front Data =294
Front Data =304
Front Data =297
Front Data =301
Front Data =307
Front Data =297
Front Data =301
Front Data =307
Front Data =294
Front Data =302
Front Data =308
Front Data =318
Front Data =295
Front Data =299
Front Data =304
Front Data =301
Front Data =295
Front Data =308
Front Data =307
Front Data =312
Front Data =317
Front Data =302
    
```

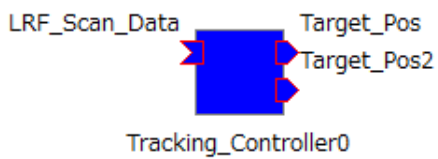


以上で URG 計測 RTC の解説は終了です。URG で距離を計測することはできたでしょうか。

4. 人追尾コントローラ RTC

4.1. 人追尾コントローラ RTC の概要

まずは人追尾コントローラ RTC の概要を以下に示します。

| RTC 名 | Tracking_Controller | | |
|-------|--|---------------|--|
| 構成 |  | | |
| | Data Port | | |
| | ポート種類 | ポート名称 | 意味 |
| | InPort | LRF_Scan_Data | URG の距離データを受信します。 |
| | OutPort | Target_Pos | パン軸モータへの角度指示値[deg]を送信します。 |
| | OutPort | Target_Pos | チルト軸モータへの角度指示値[deg]を送信します |
| | Configuration Parameter (主要なもの) | | |
| | 名称 | 初期値 | 意味 |
| | max_dist | 2000 | 計測距離の最大値を設定します。この値以上のデータは全て「0」となります。単位は mm. |
| | object_rate | 0.8 | 背景距離データに対してこの割合以下の距離が得られた箇所に物体があると認識します。 |
| | connect_size | 100 | 隣接する検出物体同士の距離がこの値以下である場合は1つの物体とします。単位は mm. |
| | lost_limit | 50 | 人追尾中この値で示すスキャン回数以上だけ連続して人の存在を見失った場合、追尾を中止して再び人を探し始めます。 |

4.2. RTC ミドルウェアを利用した動作確認

人追尾コントローラ RTC は単なるコントローラなので細かな設定は必要ありません。早速ですが動作確認に入ります。ネーミングサービスと RT System Editor はあらかじめ起動しておいてください。（操作は 1.4. ～1.5 にて解説済）







このコントローラの動作確認として以下のようにモータ先端に付けた矢印が人の方向を指し続けるというデモを行います。



- ① まずはハードウェアを構築します。といっても上図を参考にしてビニールテープ等で測域センサの上にモータを固定し、モータの出力軸に矢印となるものを取り付けるだけで構いません。矢印は EPOS 起動時に URG の前方を向くように取り付けてください。
 注意点として、人を発見しやすくするために極力測域センサの計測範囲に細かい形状の障害物が入らないようにしてください。URG の計測範囲は広いので、簡易的な対策として不必要な範囲をマスキングテープなどで覆ってもよいでしょう。テープ部分のデータは「0」となります。



- ② 次にシステムを準備します。プロジェクトページから人追尾コントローラ RTC をダウンロードして任意のフォルダに保存してください。先頭のフォルダ名は「Tracking_Controller」です。

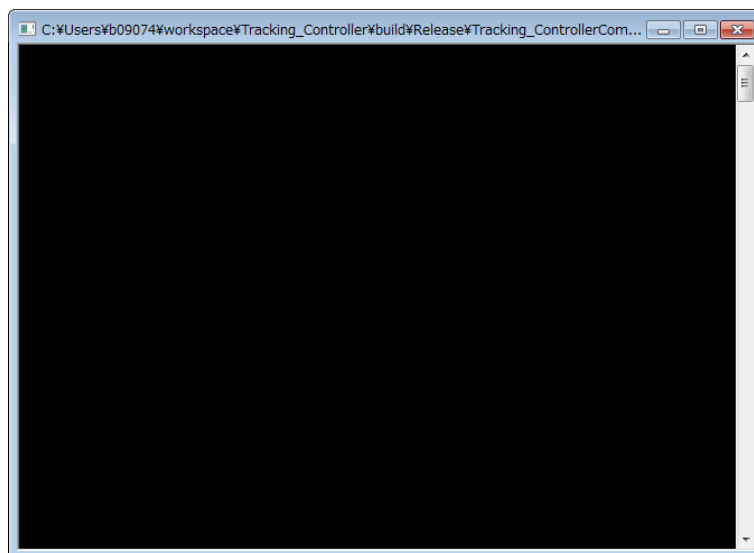
| | | | | |
|---|-----------------------|------------------|-------------|------|
|  | .metadata | 2012/11/13 16:36 | ファイル フォル... | |
|  | Maxon_motor | 2012/11/14 1:42 | ファイル フォル... | |
|  | Tracking_Controller | 2012/11/14 1:42 | ファイル フォル... | |
|  | URG_scan | 2012/11/14 3:15 | ファイル フォル... | |
|  | systemeditor0.log | 2012/11/16 5:40 | テキスト ドキュ... | 2 KB |
|  | systemeditor0.log.lck | 2012/11/16 5:19 | LCK ファイル | 0 KB |

- ③ 保存したフォルダ内を「build」「Release」と辿ると「Tracking_Controller.exe」という実行ファイルがあるのでダブルクリックして実行してください。

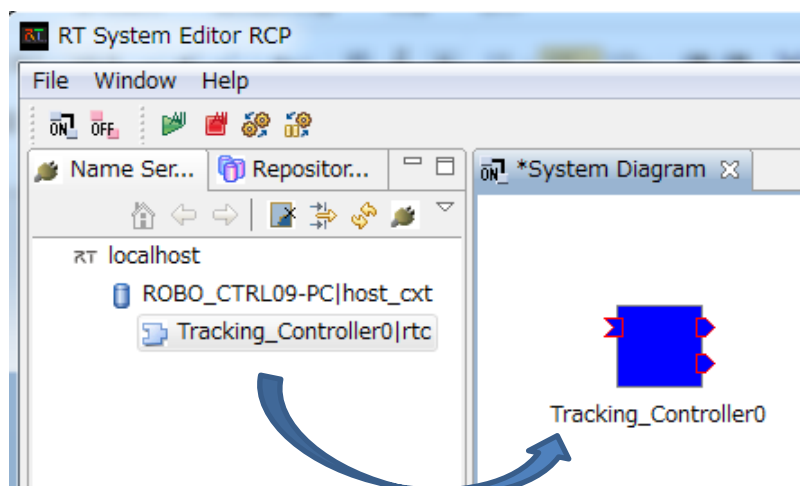
| | | |
|---|-----------------------------|-------|
|  | Tracking_Controller.dll | 82 KB |
|  | Tracking_Controller.exp | 1 KB |
|  | Tracking_Controller.lib | 2 KB |
|  | Tracking_ControllerComp.exe | 11 KB |

「Tracking_Controller.exe」
をダブルクリック

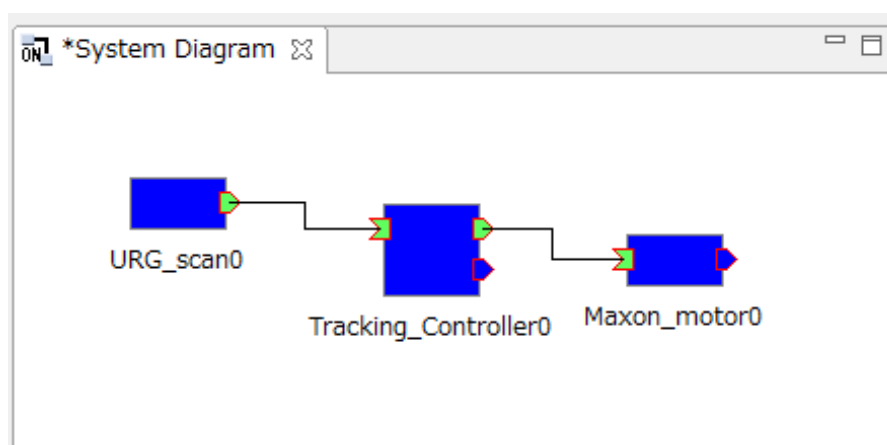
- ④ 真っ黒なウィンドウが立ち上がりますので、そのままにしておきましょう。



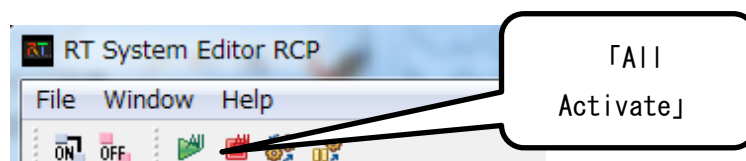
- ⑤ Tracking_Controller.exe でウィンドウを立ち上げると RT System Editor 上に起動したコンポーネントが表示されるので System Diagram 上にドラック & ドロップしましょう。



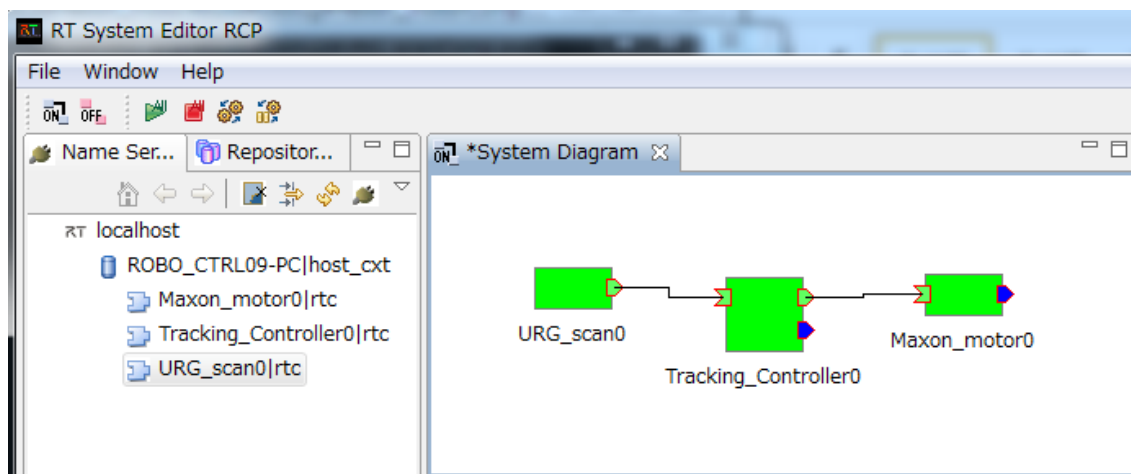
- ⑥ Tracking_Controller はコントローラですので Activate する前に、これまで設定してきたモータドライバ RTC と URG 計測 RTC を立ち上げて以下のように RTC 同士を接続してみましょう。忘れがちですが, URG 計測 RTC のコンフィギュレーションパラメータで URG の COM ポート番号を再び設定しましょう。



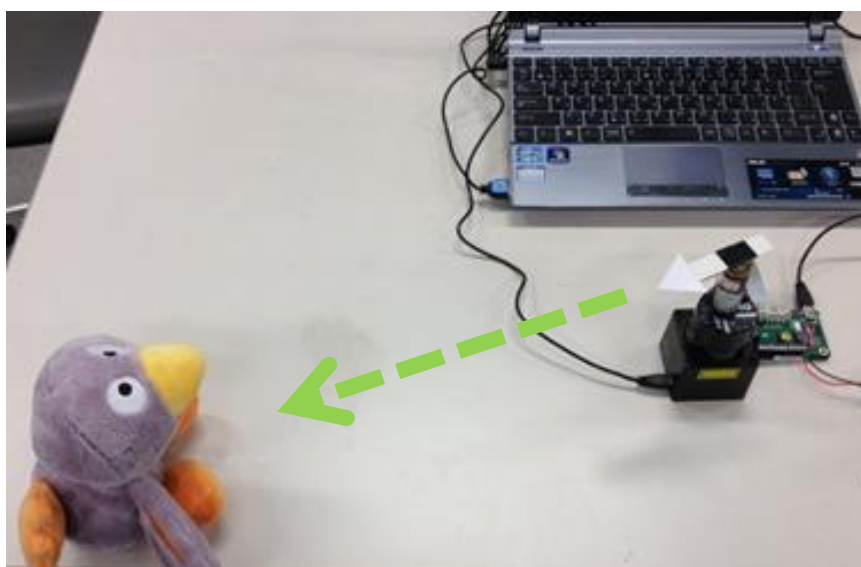
- ⑦ 以上の準備が出来たら「All Activate」のを表す緑の三角マークをクリックしてみましょう。人追尾コントローラ RTC は Activate されて最初に取得したデータを背景の環境データとして登録します。したがって, Activate した瞬間から数秒の間は自分の体を含め, URG の測域に背景となる静止物体以外を置かないでください。



- ⑧ System Diagram 上で RTC の表示が緑色に変化すれば接続成功です。赤色に変化した場合はエラーを意味します。先ほど真っ黒だったコンソール上にエラーメッセージが現れますので参考にしてみてください。



- ⑨ Activate して数秒後にモータが動き出します。URG の前を歩きまわって矢印の向きが自分を追尾していることを確認してください。



以上で人追尾コントローラ RTC の解説は終了です。パン軸のみですが、人追尾動作の実現はできたでしょうか。

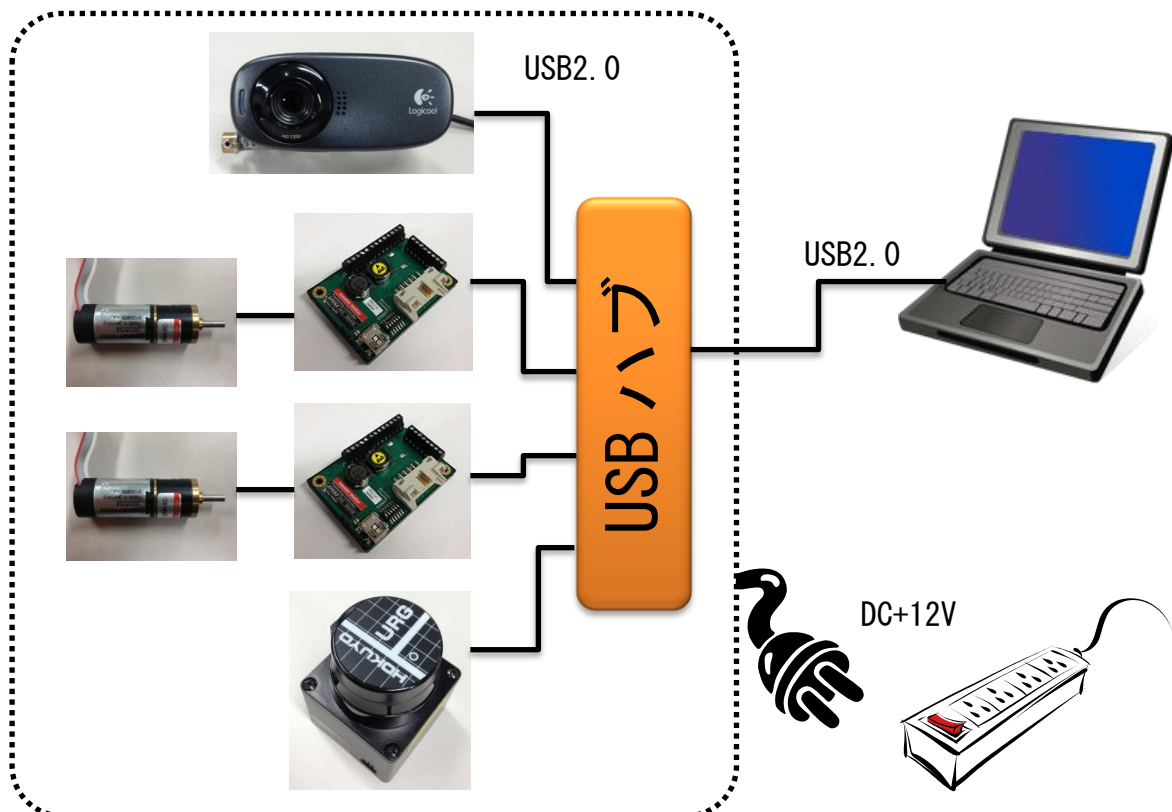
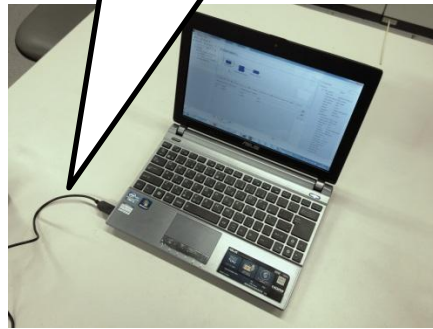
5. 人追尾カメラモジュール

5.1. システム構成

これまで解説した3種類の人追尾カメラモジュール制御 RTC と OpenRTM-aist 公式 HP 上でダウンロード可能な既存 RTC および OpenRTM-aist-1.1.0 に付属のサンプル RTC を活用して、人追尾カメラモジュールを作成することが可能です。以下にその外観とシステム構成を示します。



USB1 本で制御可能!!



5.2. 既存 RTC の再利用

既存の RTC を再利用することで人追尾カメラはさらに高機能になります。今回は s-kurihara 様が公開している「顔検出コンポーネント」を再利用することで、取得した画像から人の顔を検出してみましょう。ダウンロードは以下の URL に示す OpenRTM-aist 公式 HP から行います。

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

動作確認も行いますので、ネーミングサービスと RT System Editor はあらかじめ起動しておいてください。（操作は 1.4. ～1.5 にて解説済）

- ① URL の先では以下のような画面が表示されますのでページ下部の「ホームページ」と書かれている箇所をクリックします。



- ② 現れたページ上で「動作確認」と書かれているところが以下の画面となります。上にある「こちら」をクリックしてファイルを任意の場所にダウンロードしましょう。

動作確認

使用するコンポーネント群

DirectShowCamComp(OpenRTM-aistのサンプルに付属)
「スタート」>「すべてのプログラム」>「OpenRTM-aist 1.1」>「C++」>「components」>「opencv-rtcs」と通り、DirectShowCamComp.exeを実行します。

FlipComp(OpenRTM-aistのサンプルに付属)
「スタート」>「すべてのプログラム」>「OpenRTM-aist 1.1」>「C++」>「components」>「opencv-rtcs」と通り、FlipComp.exeを実行します。

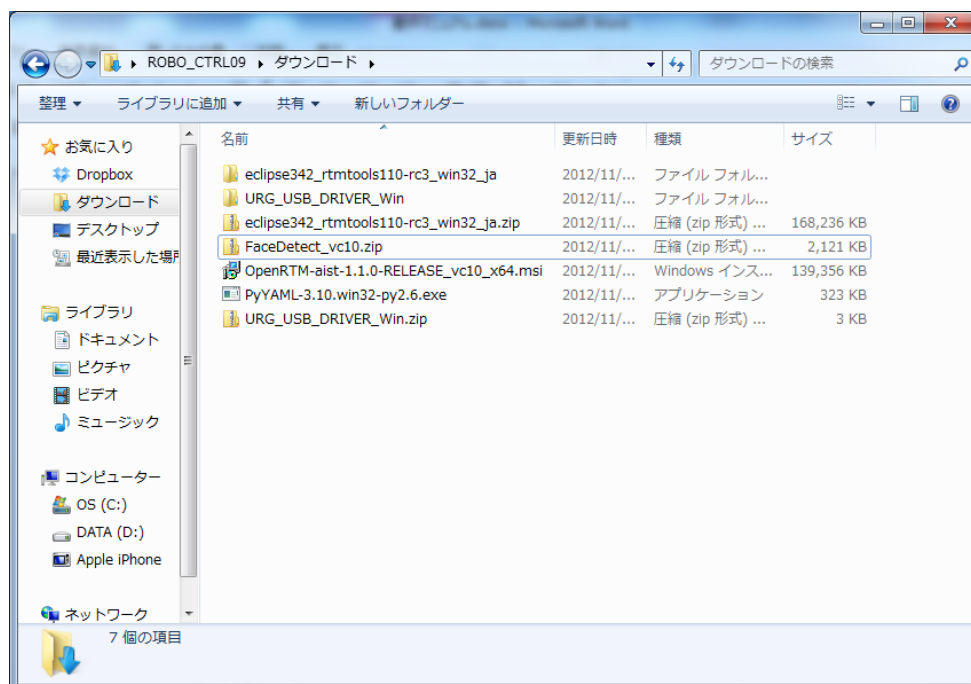
CameraViewerComp(OpenRTM-aistのサンプルに付属)
「スタート」>「すべてのプログラム」>「OpenRTM-aist 1.1」>「C++」>「components」>「opencv-rtcs」と通り、CameraViewerComp.exeを実行します。

FaceDetectComp

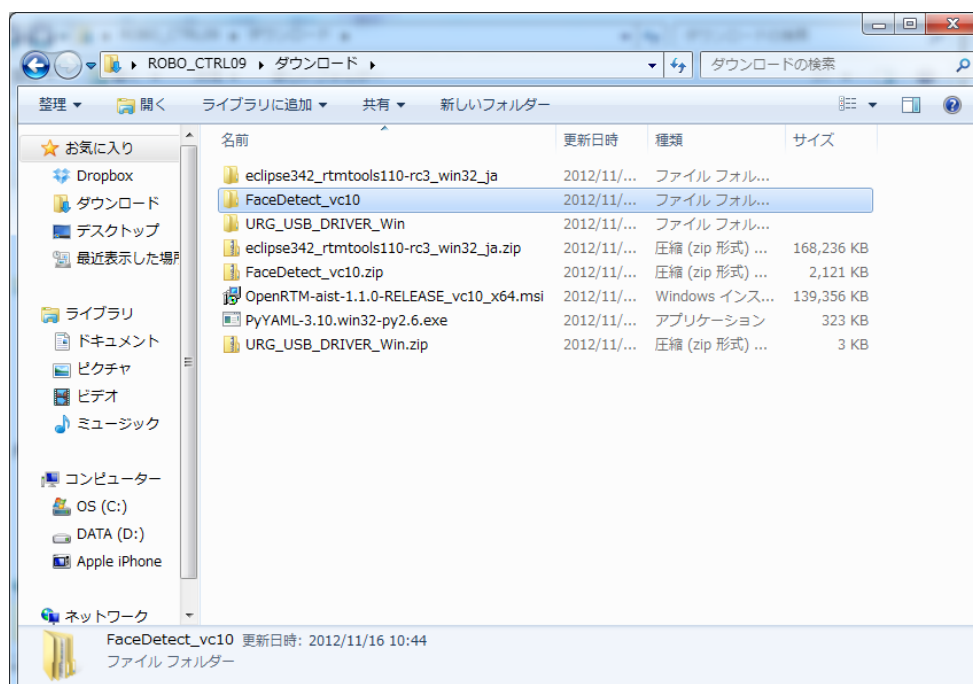
1. **こちら**からFaceDetect_vc10.zipをダウンロードし、適当なフォルダに展開します。
VC9用は**こちら**からFaceDetect_vc9.zipをダウンロードしてください。
2. 展開したフォルダ内の"build" > "Release"と通り、FaceDetectComp.exeを実行します。

「こちら」をクリック

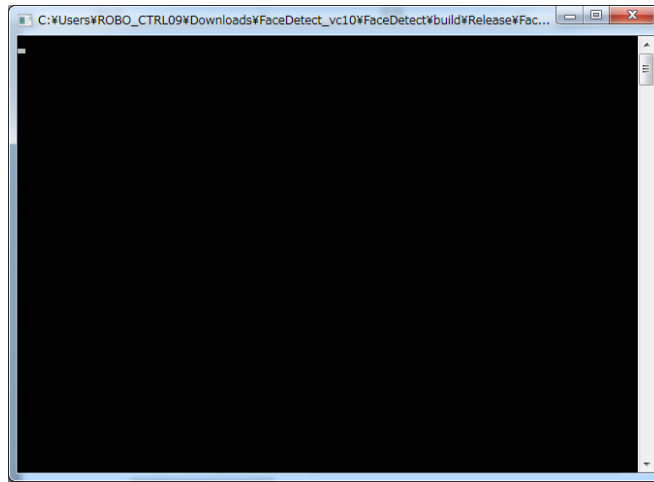
- ③ ダウンロードしたディレクトリを確認すると「FaceDetect_vc10.zip」というファイルがあるので右クリックから「すべて展開」を選択し、同じ場所に展開してください。



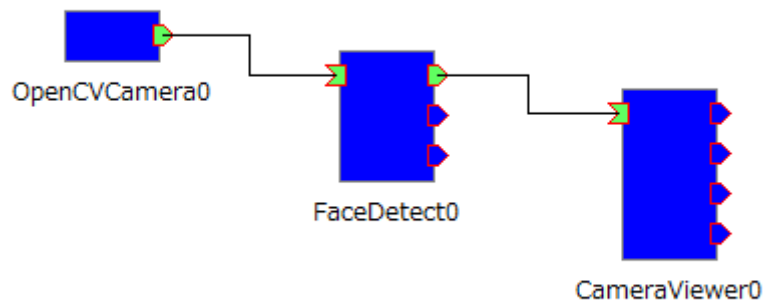
- ④ 展開したファイル「FaceDetect_vc10」が現れますので、そのファイル内を「FaceDetect」「build」「Release」と辿って「FaceDetectComp.exe」を起動しましょう。



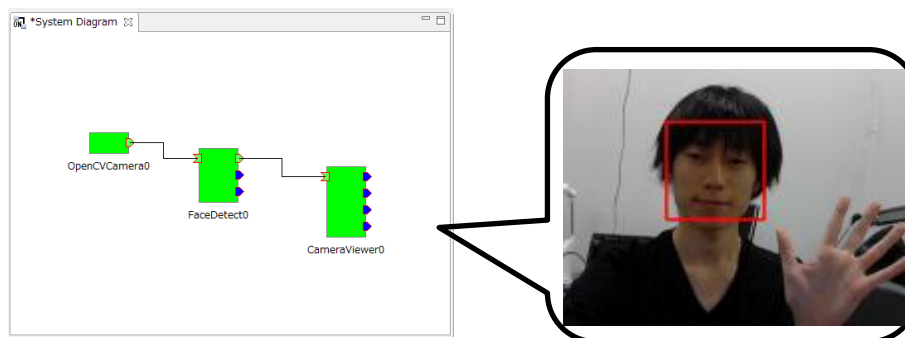
- ⑤ 真っ黒なウィンドウが立ち上がりますので、そのまましておきましょう。



- ⑥ サンプルとして付属している「CameraViewerComp」と「OpenCVCamerComp」を起動して RT System Editor 上で以下のように接続したら All Activate を実行してください。

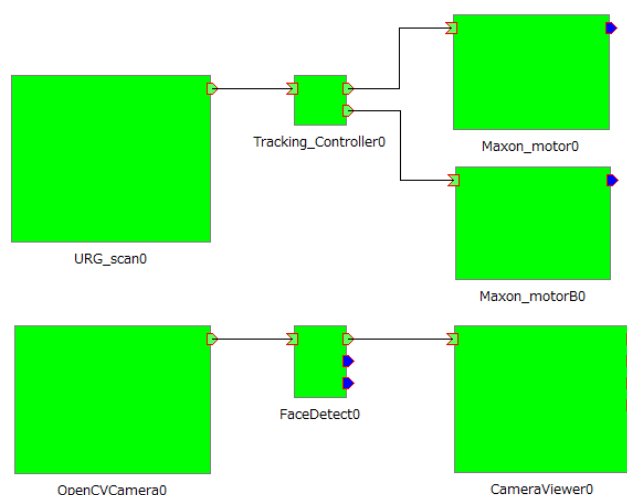


- ⑦ 全てのコンポーネントが正常に動作したら「CameraViewerComp」で立ち上がったウィンドウを確認しましょう。以下のようにリアルタイムで顔検出が行われています。赤枠で囲まれた箇所が顔と判断された部分です。



5.3. RTC の接続と実行

最後に、このマニュアルでインストールして動作確認をして解説を行ったコンポーネントを総結集してカメラモジュールを動かします。以下のように各 RTC を起動し接続してください。準備が出来たら All activate で実行します。



このシステムではモータがパン軸・チルト軸にそれぞれについているので、設定したおよその顔高さと URG によって得られた距離を基にチルト軸も制御しています。したがって、モジュールに対して前後に移動した場合でも人の顔位置が上下に見切れることなく撮影が可能です。これにより様々な位置に移動した場合でも顔検出が可能であることが確認できました。

5.4. さらに発展した利用法

チルト軸の補正

顔検出を搭載したことによってモータに搭載されたエンコーダの値と取得した画像上での顔位置から顔の高さが計測可能となるはずです。顔の高さが計測できるということは、個人の身長に合わせてチルト軸の位置補正を行い、より顔検出を容易にすることが期待できるかもしれません。

個人の同定

顔検出をさらに発展させて顔画像から個人を同定することができれば、追尾カメラモジュールの用途はさらに広がります。防犯システムや受付などが考えられますが、これを実現するためには人の顔を明確に映す必要があります。本モジュールでは人の顔付近を撮影することが可能なので個人の同定もしやすいのではないのでしょうか。

再利用における注意事項（ライセンス等）

解説の冒頭でも述べましたが、人追尾カメラモジュール制御 RTC は修正 BSD ライセンスを適応します。また、「モータドライバ RTC」と「URG 計測 RTC」はそれぞれマクソンジャパン（株）と北陽電機（株）が公開しているサンプルプログラムの一部とライブラリ利用しますので研究用途でのみ利用してください。最後に、開発者および知能機械システム研究室は本コンポーネントおよびユーザーマニュアルの利用によって生じたいかなる損害にも責任を負いかねますのでご了承ください。

参考文献・プログラム

人追尾カメラモジュール制御 RTC では以下で公開されているライブラリやサンプルプログラムの一部を利用しています。再利用させていただいた s-kur ihara 様の顔検出 RTC のダウンロードページと合わせて以下に示します。

- [1] Maxon Motor: 製品概要 サーボアンプ/位置制御 (last accessed 2012/11/19)

http://www.maxonjapan.co.jp/product_mmc.htm

- [2] 北陽電機: URG プログラミングガイド (last accessed 2012/11/19)

http://www.hokuyo-aut.co.jp/02sensor/07scanner/download/urg_programs/

- [3] RT-Middleware: 顔認識コンポーネント (last accessed 2012/11/19)

<http://www.openrtm.org/openrtm/ja/node/4629>

RT ミドルウェアの基礎を学習する上で、以下のサイトや文献を参考にしました。一覧にはありませんが、産業技術総合研究所で 2012 年に行われた「RT ミドルウェアサマーキャンプ 2012」に参加することで RT コンポーネント開発の一通りの流れをみっちり教えていただきました。

- [4] ysuga.net : RT ミドルウェア (last accessed 2012/11/19)

<http://ysuga.net/robot/rtm>

- [5] 長瀬 雅之, 中本 啓之, 池添 明宏. 「はじめてのコンポーネント指向ロボットアプリケーション開発 RT ミドルウェア超入門」. 毎日コミュニケーションズ. 2008 年.