

モジュール仕様書
緊急停止デバイス管理モジュール

Ver. 1.0.2

2011年6月27日

セグウェイコンソーシアム

京都大学

目次

改版履歴	i
目次	ii
1. はじめに	1
1. 1. 本書適用範囲	1
1. 2. 回路に関して	1
1. 3. 関連モジュール	2
1. 4. 本書の対象者	2
2. RTC 詳細	3
2. 1. 概要	3
2. 2. 使用環境設定	3
2. 3. DeviceControlComp	4
2. 3. 1. 機能概要	4
2. 3. 2. ポート情報	4
2. 4. ButtonConverterComp	5
2. 4. 1. 機能概要	5
2. 4. 2. ポート情報	5
3. モジュール使用方法	7
3. 1. ハードウェアの準備, 設定	7
3. 2. 環境の整備	7
3. 3. コンパイル方法	8
3. 4. 起動手順・使用方法	8
3. 4. 1. その他ハードウェア準備	8
3. 4. 2. 起動手順	8
3. 4. 3. モジュール構成例	9
3. 4. 4. 使用方法	10
3. 4. 5. 注意事項	10

1. はじめに

1. 1. 本書適用範囲

本書では、ダックス技研社が販売しているUSB接続デジタル入出力基板を用い、緊急停止スイッチを管理するモジュールの構成説明及び使用手順を記述している。また、パトライト社製のパトランプの点灯の制御も同時に行っており、安全性向上を目的に開発したモジュールである。本RTCで使用する部品を下記表に示す。

表 1 必要な部品

型番	メーカー	HP
Dacs-2600	ダックス技研	http://www.dacs-giken.co.jp/d2600.html
HW1X-BV401R	IDEC	http://www.idec.com/jpja/products/Catalogs/SafetyProducts/HW1X/index.html
LCE-302-AFB W-RYG	パトライト	http://www.patlite.co.jp/product/detail.php?i=658

1. 2. 回路に関して

本RTCを利用するには、USB接続デジタル入出力基板に非常停止スイッチとパトライトを接続する必要がある。その接続を行うには、新たに回路を作成する必要がある。回路図を下記に示すので、使用者はまず回路図を参考に回路を作成されたい。(回路図は本書と同じフォルダにも格納している)。

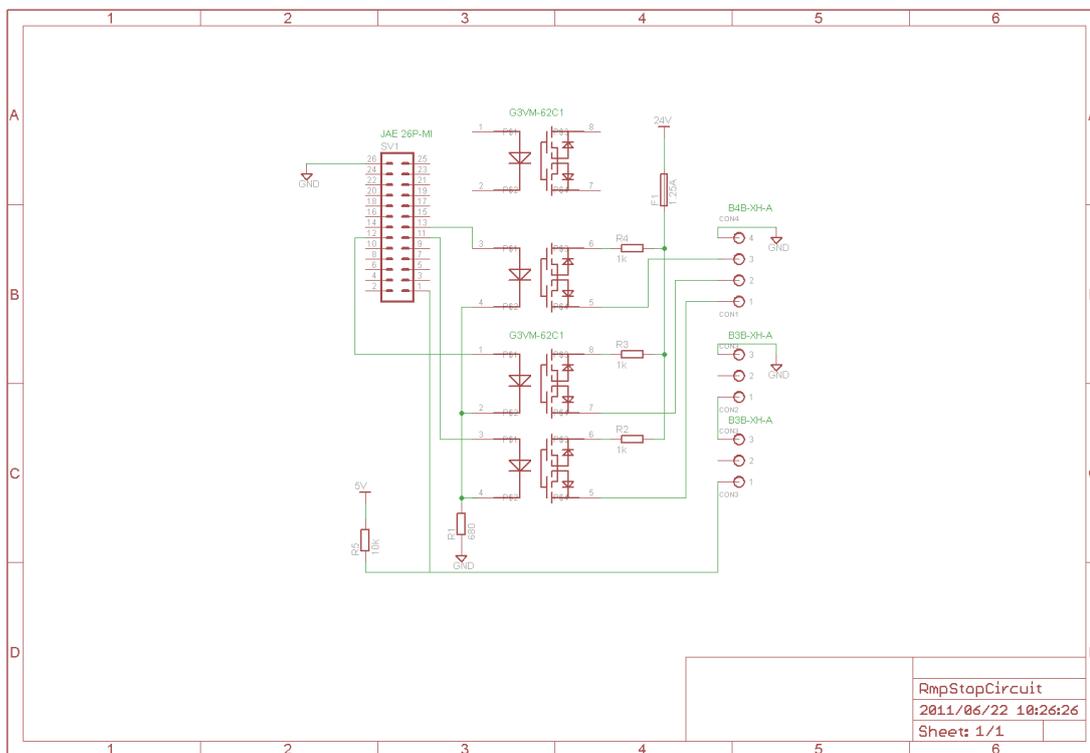


図 1 回路図 1

2. RTC 詳細

2. 1. 概要

本知能モジュールは、ダックス技研の USB 接続デジタル入出力基板 DACS-2600(参考 HP:<http://www.dacs-giken.co.jp/d2600.html>)を用いてパトライトや緊急停止スイッチデバイスを監視するモジュールである。詳細な接続図を図 3 に示す。図にあるように DACS-2600 には、自律と融合モジュールからは操縦モード/自律モードの情報が USB を介して送られてくる。また、緊急停止スイッチとも繋がっており、スイッチが押されたかどうか(導通・断線)もこの基板では監視している。DACS-2600 からのアウトプットとして、操縦モード(オレンジ)/自律モード(赤)/緊急停止モード(緑)かのシリアルデータをパトランプに送っている。また、緊急停止スイッチが押された場合は、自律と操作の融合モジュールに緊急停止情報を送り、ソフトウェア的に速度を 0 にする役割をしている。

本モジュールと関連デバイスを使用することで、現在どんなモードで走行しているのかをパトランプで判断することが出来る。また緊急停止スイッチが押された場合は、ソフトウェア的に強制的に停止させることが出来る。なお、本モジュールには、ゲームパッドからのボタン情報を変換して、操作と自律の切り替え情報を融合モジュールに送る ButtonConverter モジュールも含めている。

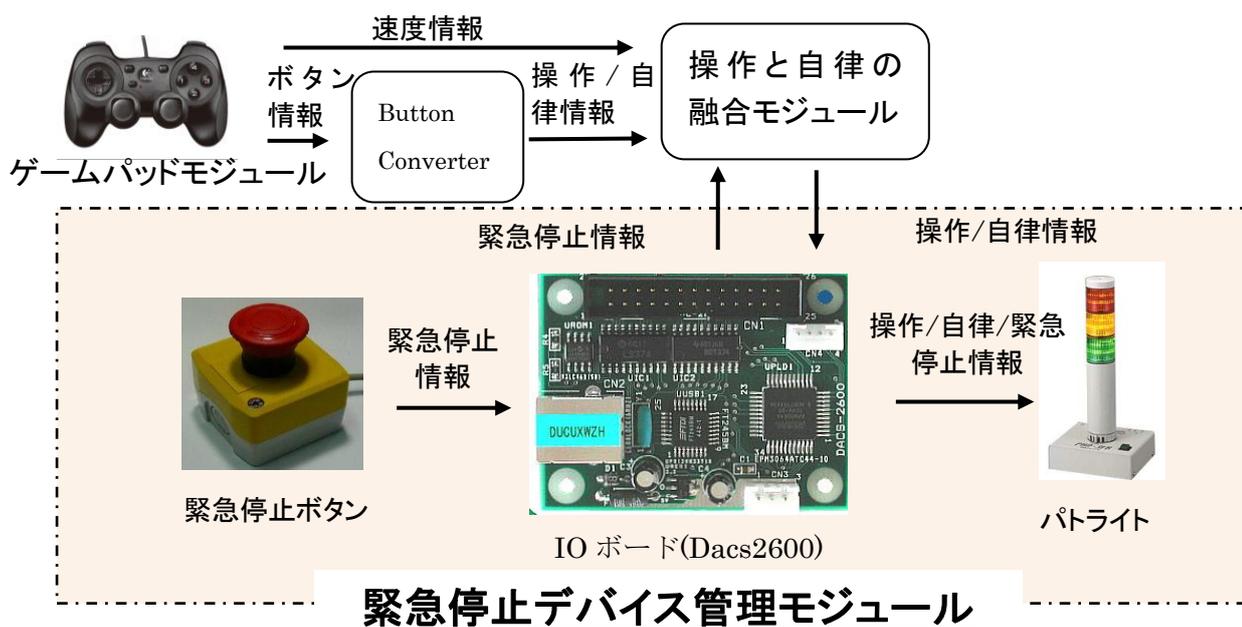


図3 緊急停止デバイス管理モジュールシステム構成図

2. 2. 使用環境設定

本書で述べる RTM システムは以下に述べる環境での動作確認を行っている。なお、本 RTC はオープンソースで公開しているので、再コンパイルすれば他環境でも動作可能であると考えられる。

表 2 確認済み環境設定

動作 OS	Windows 7 32 bit / Ubuntu 10.04 32bit
開発言語	C++
コンパイラ	Visual Studio (Visual C++ 2008 Express Edition) / gcc
RT ミドルウェア/バージョン	OpenRTM-aist-C++-1.0.0 Release
依存パッケージ	boost 1.40 以降 (DACS-2600 使用には付属ソフトのインストールが必要.)
モジュール操作	RT System Editor
本書内で使用した PC	Vaio-Z(VGN-Z90NS)

2. 3. DeviceControlComp

この節では、図 3 の緊急停止スイッチとパトランプを制御する緊急停止デバイス管理に対応するモジュールである DeviceControl モジュール (図 4 参照) についての機能概要、詳細について述べる。



図 4 DeviceControl モジュールの入出力ポート

2. 3. 1. 機能概要

この DeviceControl モジュールは、大きく 2 つの役割がある。1 つ目は、緊急停止機能である。USB 接続デジタル入出力基板に取り付けられた緊急停止スイッチを押すことでソフトウェア側の速度を強制的に 0 にし、それがロボットに送られるという機能がある。2 つ目は状態表示である。ゲームパッドからの自律/操作の切り替え情報を取得し、同様の USB 接続デジタル入出力基板に接続されたパトランプの表示灯に自律/操作/非常停止の状態を表示させる機能がある。

2. 3. 2. ポート情報

○データポート(InPort)

表 3 InPort 一覧

ポート名	型	説明
lightColor (状態)	TimedLong	表示灯の色を変更するための指令値が入力される 1:緊急停止(緑), 2: 操作(オレンジ), 3:自律(赤)

○データポート(OutPort)

表4 OutPort 一覧

ポート名	型	説明
emoState (緊急停止)	TimedBoolean	緊急停止の指令を出力(true で非常停止状態)

○Configuration

表5 Configuration 一覧

ConfigurationSet	型	Default Value	説明
default	portname (string 型)	COM1(または) /dev/ttyUSB1	デバイスマネージャや dmesg の結果を参照しポートを設定すること。

2. 4. ButtonConverterComp

この節では、図3に示した ButtonConverterComp について説明する。

2. 4. 1. 機能概要

ButtonConverter はゲームパッドから受け取った、ボタン情報(TimedULong)をもとに、操作モード(TimedBoolean)を生成し出力するためのものである(図5参照)。どのボタンを押せば操作モードになるか、自律走行モードになるかはコンフィギュレーションにより設定が可能となっている。

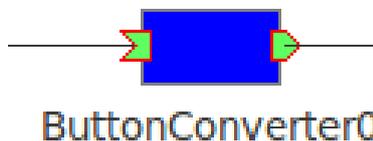


図5 ButtonConverter モジュールの入出力ポート

2. 4. 2. ポート情報

○データポート(InPort)

表6 InPort 一覧

ポート名	型	説明
button (ボタン情報)	TimedULong	ゲームパッドモジュールから出力されたボタン情報(詳細はゲームパッドモジュールのマニュアルを参照のこと)

○データポート(OutPort)

表7 OutPort 一覧

ポート名	型	説明
------	---	----

ポート名	型	説明
doAuto (操作モード)	TimedBoolean	操作モードを設定する(true で自律走行モード)

○Configuration

表 8 Configuration 一覧

ConfigurationSet	型	Default Value	説明
buttonNum_Manual	long	1	操作モードにするボタン番号
buttonNum_Auto	long	2	自律走行モードにするボタン番号

3. モジュール使用方法

3. 1. ハードウェアの準備, 設定

図6のように, IO ボード, パトライト, 非常停止スイッチを配置する.

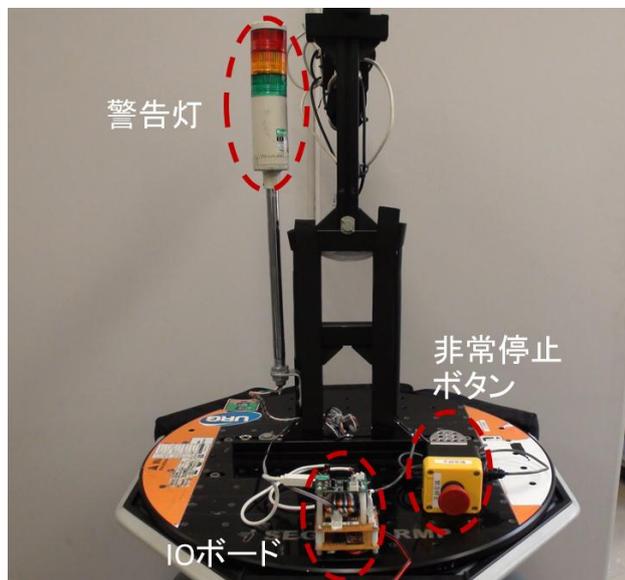


図6 各デバイスの配置例

3. 2. 環境の整備

○OpenRTM-aist のインストール

モジュールを使用するには OpenRTM-aist 1.0.0 が動作する環境を整える必要がある。以下の手順で使用する環境に応じてインストールを行う。なお、環境変数等は下記に従い行えばよく、追加で設定する必要のあるものはない。

- ・ C++

【ダウンロード】 <http://www.openrtm.org/openrtm/ja/node/849>

【マニュアル】 <http://www.openrtm.org/openrtm/ja/node/999>

○RT SystemEditor をインストール

【ダウンロード】 <http://www.openrtm.org/openrtm/ja/node/941>

【マニュアル】 <http://www.openrtm.org/openrtm/ja/node/676>

○依存ライブラリのインストール

- boost

下記サイトから使用する OS にあったパッケージをダウンロードし、マニュアルに従ってインストールする。

<http://www.boost.org/>

- DACS2600 U S B接続デジタル入出力基板依存ライブラリ

USB接続デジタル入出力基板は依存ライブラリがある。製品と同梱されている CD-R よりインストールする。

製品情報およびインストール方法はダックス技研株式会社ホームページ：

<http://www.dacs-giken.co.jp/d2600.html>

もしくは取扱説明書：

<http://www.dacs-giken.co.jp/DA26manu.pdf>

を参照のこと。

3. 3. コンパイル方法

・ Windows 環境(VC++2008)でのコンパイル方法

1. コンパイルしたいモジュール名のフォルダに移動する。
2. copyprops.bat を実行する
3. (モジュール名)_vc9.sln をダブルクリックして、VC++のソリューションを開く。
4. Ctrl+F5 を押してビルドする。
5. /(モジュール名)Comp フォルダに(モジュール名)Comp.exe が生成されているので、適当な場所にコピーする。

以上。

・ Ubuntu 10.04 でのコンパイル方法

1. コンパイルしたいモジュール名のフォルダにある src フォルダに移動する。
2. Terminal で make を実行する。
3. (モジュール名)Comp が生成されているので、適当な場所にコピーする。

以上。

3. 4. 起動手順・使用方法

この節では、具体的な手順を説明する。なお、必要な DeviceControlComp, ButtonConverterComp を含め、Comp ファイルと rtc.conf は同一のフォルダにあるものとする。

3. 4. 1. その他ハードウェア準備

今回は例として、ロボット状態提示用にゲームパッド(Logitech社)を使用する。詳細はゲームパッド説明書に記述のため、ここでは省略する。

3. 4. 2. 起動手順

(1) ネームサーバの起動

- ・ コマンドプロンプト画面で「rtm-naming」と入力し、ネームサーバを立ち上げる。

注) Ubuntu の場合には、デフォルトで立ち上がっているネームサーバを利用すればよい。もし、

独自の設定で使用したい場合にはまず「sudo killall omniName」と打ち込み、ネームサーバを落とし、そののちネームサーバを立ち上げる必要がある。

(2) rtc.conf の調整

- ・同じ PC でネームサーバを立ち上げているため、corba.nameservers: 127.0.0.1:2809 のままでよい。もし、別 PC なら IP アドレスを変更する。

(3) RT System Editor を起動する。

(4) 各 RTC の起動

DeviceControlComp, ButtonConverter を実行. 今回は例として同コンソーシアムから公開しているゲームパッド用モジュール (GamePad) と自律と操作の融合モジュール (CommandSelectorComp) を起動する. ただし, それぞれの説明書を読み, 環境を事前に整えておくこと. 環境が整ってないままモジュールを **Active** にするとエラーが起きる.

(5) RT SystemEditor 上でのモジュール接続

- ・次に示す図 7 のようにモジュールを接続する.

3. 4. 3. モジュール構成例

モジュールの構成の例を図 7 に示す. DeviceControlComp はシリアルポートの設定が出来ていれば, 単体だけでもエラーは起こらず起動できる. なお, ここでは簡単化のため CommandSelector の上から 2, 3 番目の入力ポートの両方にゲームパッドからの出力を接続している. 3 番目のポートが自律走行のための速度指令値を入力するためのポートであり, 自律走行用の適切なモジュールを接続し, 移動ロボットを CommandSelectorComp の 1 番目の出力ポートに接続すれば操作と自律走行を切り替えながら動かすことができる.

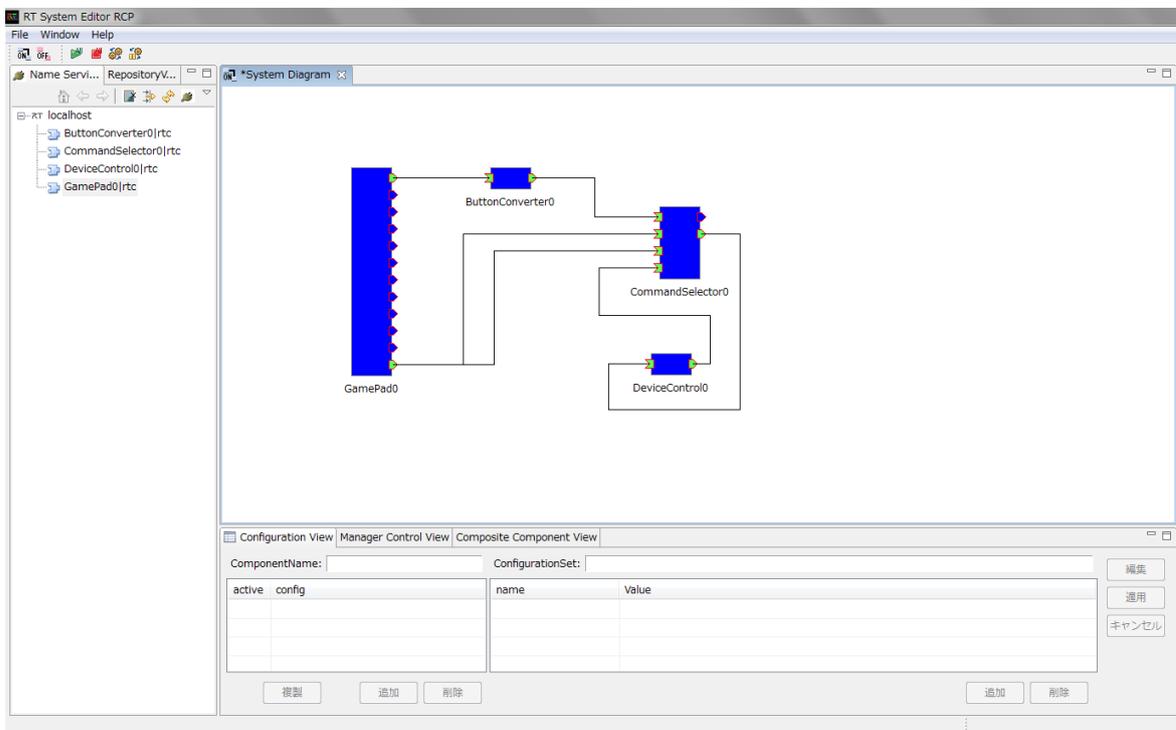


図 7 モジュール構成例

3. 4. 4. 使用方法

- (1) デバイス系の電源をつけ、USB を接続する。
 - (ア) USB 接続デジタル入出力ボードに電源が入っていることを確認し、USB を接続する
 - (イ) GamePad の USB を接続する
- (2) DeviceControl モジュールを起動し、Configuration を変更する。
 - デバイスマネージャ/dmesg を参考に portname のポート名があっているか確認し変更する
- (3) ButtonConverter, CommandSelector, Gamepad モジュールを起動する。
- (4) 全てのモジュールをアクティブにする。
 - ※エラーの場合はモジュールが赤くなる。
- (5) 成功の場合は、全てがアクティブの黄緑色になり表示灯が点灯する。ゲームパッドの“1”のボタンを押すと操作モードとなり表示灯はオレンジ色が点灯し、ジョイスティックを傾けない限りロボットは停止する。“2”のボタンを押すと自律走行モードに切り替わるため、表示灯は黄色になり（自律走行モジュールを接続している場合）自律で走行を始める。緊急停止ボタンを押すと緑色ランプが点灯し、停止する。緊急停止ボタンのロックを解除しない限り、ジョイスティックを傾けても走行することはない。

3. 4. 5. 注意事項

- DeviceControlComp 単体でも起動可能。しかし、操作/自律の状態入力は出来ない。
- アクティブの際、一つでもエラーが起こり、モジュールが赤くなるとモジュールは全て起動しなおす必要がある。
- その他のデバイス依存のライブラリはインストールしておくこと。
- ロボットの最大速度を調整する機能はないため、ロボットの最大速度、ゲームパッドの出力の最大速度、自律移動モジュールの最大速度を確認しておくこと。
- ゲームパッドは「PlayStation のコントローラと USB 変換」とのセットの場合は Logicool, Logitech のゲームパッドの機種とボタン配置が異なるため使用できない可能性がある。
- 本モジュールの動作確認用の動画を参照することを勧めます。