

1 RTC 概要

E1 (T54) は6軸+把持1軸のアームを左右に2つ有する、クローラ型ロボットである。ゲームコントローラ等と接続し、E1(T54 Fig. 1)操作用コマンドを出力する。Windows上で動作。Linux未検証。RTC_GameController_Win・RTC_E1_GUIとの接続に対応。

E1 操作のためのRTC接続法・操作法に関してはパッケージマニュアルE1_Systemを参照のこと。



Fig. 1 T54

1 開発環境

OS: Windows7 Pro 32bit

RTM: OpenRTM-aist C++ 1.1.1-RELEASE

開発環境 : Visual Studio Express 2013 for Windows Desktop

2 ビルド方法・実行方法

*ビルド方法

- 任意の場所に **RTC_E1Motion_Manual** フォルダを置く
- CMake を用いて、**Configure・Generate**
- 作成されたプロジェクトファイル(.sln)を開きビルド
- 実行ファイルが作成される

*実行方法

- **RTC_E1Motion_Manual** を **Activate**。
実行中は現在の操作モード等を端末上に表示する。

3 RTC 説明

RTC_E1Motion_Manual のコンポーネントを Fig. 2 に示す。本コンポーネントの入力は3つ、出力は1つである。本 RTC の入力ポートを Table 1 に示す。

1つ目の入力である **Controller_Type** は接続されたデバイス名を取得するものであり、プログラム内で接続されたコントローラが何であるか判断するために使用している。

2つ目の入力である **Button** はゲームコントローラのボタン押下状態を **TimedULong** 型で入力する。詳細は 5 項参照のこと。

3つ目の入力である **Analog** はゲームコントローラのスティックや一部のボタンから取得したアナログ情報を **TimedDoubleSeq** 型で入力する。詳細は 6 項参照のこと。



Fig. 2 **RTC_E1Motion_Manual**

Table 1 InPort List

名称	データ型	概要
Controller_Type	TimedString	接続デバイス名
Button	TimedULong	ボタン押下状態
Analog	TimedDoubleSeq	アナログ情報

本 RTC の出力ポートを Table 2 に示す。出力は TimedString 型の Command である。
 入力ポート Button、Analog の情報から E1 操作の Command を生成する。

Table 2 InPort List

名称	データ型	概要
Command	TimedString	T54 操作 Command

4 Button Port

入力ポートは 16 進数 4 桁 (16bit) となっており、その対応表を Table 3 に示す。使用可能なボタン数は 14 であり、それ以上のボタン数の押下状態は無視される。

E1 では PS4 コントローラの使用を想定している。PS4 コントローラには計 14 個のボタンがあり、ボタンを押した際に 1 話した際に 0 とすることで 4 桁の 16 進数で押下状態を表現する。ボタン番号 1~4 の押下状態が出力の 1 桁目に対応している。以後ボタン番号 4 つ毎に出力の 2、3、4 桁目に対応している。

PS4 コントローラ以外のコントローラでもこの対応表に沿った入力を作成することで E1 のボタン操作が可能となる。

本 RTC 内では 14 ボタン全ての値を使用する。

Table 3 Button

ボタン番号	PS4	ボタン押下状態	Button (出力)
1	□	ボタンが押された際は1 押されていない場合は0	16進数1桁目
2	×		
3	○		
4	△		
5	L1		16進数2桁目
6	R1		
7	L2		
8	R2		
9	SHARE		16進数3桁目
10	OPTION		
11	L3		
12	R3		
13	PS		16進数4桁目
14	タッチパッドボタン		
15	-		
16	-		

5 Analog Port

入力ポート Analog は配列数 8 つの TimedDoubleSeq 型のデータである。各配列の対応表を Table 4 に示す。

PS4 コントローラを例に挙げると、data[0],data[1]には十字キーが割り当てられている。十字キー↑を押した際に data[0]は-1 を出力し、↓を押した際には data[0]は 1 を出力する。十字キー←を押した際は data[1]は-1 を出力し、→を押した際には data[1]は 1 を出力する。キーを押していない場合は data[0], data[1]ともに 0 を出力する。data[2]~data[5]には左スティック、右スティックの傾きに応じて-1~1 の範囲でアナログに値を出力する。スティックを倒していない場合（デフォルト）では 0 を出力する。data[6], data[7]は L2 ボタンと R2 ボタンが対応する。それぞれのボタンの押し込み具合に応じて-1~1 の範囲でアナログに値を出力する。ボタンを押していない場合-1 が出力される。

本 RTC 内では data[0]~data[5]の値を使用する。

Table 4 Analog

PS4	対応出力配列	デフォルト値	値の範囲
十字キー↑	data[0]	0	↑キー押下時：-1 ↓キー押下時：1 出力は-1,0,1の3種
十字キー↓			
十字キー←	data[1]	0	←キー押下時：-1 →キー押下時：1 出力は-1,0,1の3種
十字キー→			
左スティック 前後方向	data[2]	0	左スティックの前後方向の傾きに応じて -1~1の範囲で変化 ↑方向最大傾き時：-1 ↓方向最大傾き時：1
左スティック 左右方向	data[3]	0	左スティックの左右方向の傾きに応じて -1~1の範囲で変化 ←方向最大傾き時：-1 →方向最大傾き時：1
右スティック 前後方向	data[4]	0	右スティックの前後方向の傾きに応じて -1~1の範囲で変化 ↑方向最大傾き時：-1 ↓方向最大傾き時：1
右スティック 左右方向	data[5]	0	右スティックの左右方向の傾きに応じて -1~1の範囲で変化 ←方向最大傾き時：-1 →方向最大傾き時：1
L2	data[6]	-1	L2の押し込み具合により -1~1の範囲で変化
R2	data[7]	-1	R2の押し込み具合により -1~1の範囲で変化

6 操作とボタン・アナログ値の対応

本 RTC は機体の前後進、旋回を行う走行モードとアームを動作させるアームモードの2つのモードを有する。まず、各モード共通動作項目は走行モード・アームモードの切り替え、機体前面についたブレードの上げ下げや、照明、指示器、アクセルのアップダウン等の各種機能のオンオフである。各モード共通で動作可能な部分について、各ボタン・アナログ値と操作方法との対応を Table 5 に示す。表内ボタン番号・アナログ値は4項・5項のボタン番号・アナログ値と同義である。頭照明以降のものはボタン・アナログ値の組み合わせでオンとなる。

モメンタリ表記のある機能は該当するボタン（アナログ含む）を押している際のみオンとなる。モード切替え、アクセルアップダウンに関しては押し続けても状態は1度のみ変化する。押し続けることでモードが変化し続けるといった状態にはならない。オルタネイト表記のある機能は該当するボタン（アナログ含む）を押した際に機能のオンとなり、ボタン（アナログ含む）を離した後も状態を保持する。もう一度ボタン（アナログ含む）を押した際にオフとなる。

bit オフにする際の注意点として Button[8]または Button[7]を組み合わせた機能の場合（頭照明以降）、Button[8](Button[7])を0とする前に組み合わせたもう一方の物を0としてから Button[8] (Button[7])を0としなければならない。例えば、前面照明の場合 Button[4]を0としてから Button[8]を0とする。組み合わせたアナログ値に関しても同様である。

Table 5 Function Common

機能	ボタン番号	アナログ値	備考
モード切替	Button[14]	-	モメンタリ
ブレード上昇	Button[9]	-	モメンタリ
ブレード下降	Button[10]	-	モメンタリ
頭照明	Button[8]	data[0]=-1	オルタネイト
左腕照明	Button[8]	data[1]=-1	オルタネイト
右腕照明	Button[8]	data[1]=1	オルタネイト
前面照明	Button[4] Button[8]	-	オルタネイト
後面照明	Button[2] Button[8]	-	オルタネイト
左側面照明	Button[1] Button[8]	-	オルタネイト
右側面照明	Button[3] Button[8]	-	オルタネイト
ホーン	Button[8]	data[0]=1	モメンタリ
左指示器	Button[5] Button[7]	-	オルタネイト
右指示器	Button[6] Button[7]	-	オルタネイト
ハザード	Button[3] Button[7]	-	モメンタリ
アクセルアップ	Button[4] Button[7]	-	モメンタリ
アクセルダウン	Button[2] Button[7]	-	モメンタリ
走行切り替え	Button[1] Button[7]	-	モメンタリ

走行モード時は前述の機能操作に加え、機体の走行・本体旋回が可能となる。RTC 起動時のデフォルトはこのモードである。走行モードの機能はアナログ値のみで操作する。アナログ値と機能の対応を Table 6 に示す。対応するアナログ値が 0 である場合は停止する。アナログ値の data[2] の値が 0 未満、-1 以上、かつ data[3] の値が -0.5 より大きく 0.5 未満である際に前進する。後進、左右旋回についても同義である。本体左右旋回時は前後進と左右旋回が不可能となる

Table 6 Drive Mode

機能	アナログ値	備考
前進	data[2]	data[2] : 0~1 -0.5 < data[3] < 0.5
後進	data[2]	data[2] : 0~1 -0.5 < data[3] < 0.5
右旋回	data[3]	data[3] : 0~1 -0.5 < data[2] < 0.5
左旋回	data[3]	data[3] : 0~1 -0.5 < data[2] < 0.5
本体右旋回	data[5]	data[5] : 0~1
本体左旋回	data[5]	data[5] : 0~1

アームモード時は前述の機能操作と、アームの操作が可能である。機体の走行・本体旋回は不可能となる。走行モード時にモード切替えボタン (button[14]) を押すことでこのモードとなる。アームモードの機能はアーム軸指定とアナログ値の組み合わせで操作する。ボタン番号・アナログ値と機能の対応を Table 7 に示す。

左右の腕には対応するボタンやアナログ値が割り当てられており、操作したい軸に対応したアナログ値やボタンを表記の値とすることで軸の指定が可能である (ボタンの場合は対応 bit を 1 にする)。対応するアナログ値が 0 である場合は停止する。アナログ値が 0 未満、-1 以上の際 CW 回転し、0 より大きく 1 以下の場合 CCW 回転となる。

Table 7 Arm Mode

機能	軸指定	アナログ値	備考
左腕第1関節	data[1] =1	data[2]	data[2] : 0~-1 CW data[2] : 0~1 CCW
左腕第2関節	data[0] =1	data[2]	
左腕第3関節	data[1] =-1	data[2]	
左腕第4関節	data[0] =-1	data[2]	
左腕第5関節	Button[5]	data[2]	
左腕第6関節	Button[7]	data[2]	
左腕第7関節	Button[12]	data[2]	
右腕第1関節	Button[1]	data[3]	data[3] : 0~-1 CW data[3] : 0~1 CCW
右腕第2関節	Button[2]	data[3]	
右腕第3関節	Button[3]	data[3]	
右腕第4関節	Button[4]	data[3]	
右腕第5関節	Button[6]	data[3]	
右腕第6関節	Button[8]	data[3]	
右腕第7関節	Button[11]	data[3]	