

# RTC\_GameController 概要説明

2016/3/30

株式会社アイザック

## 1. RTC 概要

PS4 ゲームコントローラと接続し、ボタン押下状態とアナログ情報を出力する RTC。  
Linux のみで動作。Windows 未対応。現状 PS4 コントローラのみ対応。他のコントローラでは未検証。PS4 コントローラ以外の接続ではデータ出力しない。本コンポーネントで他のコントローラを接続して出力する際には RTC\_GameController.cpp 内 25 行目”#define DEV\_JUD”をコメントアウトしてビルドする。

## 2. 開発環境

OS: Ubuntu14.04

RTM: OpenRTM-aist C++ 1.1.1-RELEASE

## 3. ビルド方法・実行方法

\*ビルド方法

- 任意の場所に RTC\_GameController フォルダを置き RTC\_GameController に入る。  
\$ cd RTC\_GameController
- build フォルダを作成 build フォルダに入る。  
\$ mkdir build && cd build
- cmake 後に make。  
\$ cmake ../  
\$ make
- build フォルダ内の src フォルダ内に実行ファイル RTC\_GameControllerComp が作られる。

\*実行方法

- RTC\_GameControllerComp の Activate 前に PS4 コントローラが接続されていることを確認する。接続先のデフォルトは/dev/input/js0 である。  
(コンフィギュレーションパラメータで変更可)  
以下コマンドを実行し、js0 が存在することを確認。  
\$ ls /dev/input

- RTC\_GameControllerComp を Activate。  
 実行中はボタン押下状態、アナログ状態を端末上に表示する。  
 接続先が存在しないまま Activate した場合には ERROR を返す。

#### 4. RTC 説明

RTC\_GameController のコンポーネントを Fig.1 に示す。本コンポーネントの出力は 3 つである。Table1 に示す。



Fig.1 RTC\_GameController

Table 1 Out Port 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

1 つ目の出力である Controller\_Type は接続されたデバイス名を取得し TimedString 型で出力する。PS4 コントローラを接続した場合, "Wireless Controller" と出力される。このデバイス名はプログラム内で接続されたコントローラが PS4 コントローラであるか判断するために使用している。

2 つ目の出力である Button はゲームコントローラのボタン押下状態を TimedULong 型で出力する。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 進数を生成し出力データとする。その対応表を Table 2 に示す。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。

Table 2 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	-		
		0	

Table 3 Button Example

ボタン番号	PS4	ボタン押下状態	Button (出力)
1	□	0	16進数1桁目: 8
2	×	0	
3	○	0	
4	△	1	
5	L1	1	16進数2桁目: f
6	R1	1	
7	L2	1	
8	R2	1	
9	SHARE	0	16進数3桁目: 0
10	OPTION	0	
11	L3	0	
12	R3	0	
13	PS	0	16進数4桁目: 2
14	タッチパッドボタン	1	
15	-	0	
16	-	0	

例として△、L1、L2、R1、R2、タッチパッドボタンを押した際の出力対応表を Table 3 に示す。出力の16進数1桁目に対応するボタンで押されているものは△ボタンのみであるため押下状態は”1000”となり、16進数での出力1桁目は”8”となる。2桁目の押下状態はL1、L2、R1、R2ボタンが押されているため、”1111”である。よって出力2桁目は”f”となる。3桁目に対応するボタンは押されていないため押下状態は”0000”であり、出力3桁目は”0”となる。4桁目に対応するボタンの中ではタッチパッドボタンが押されているため押下状態は”0100”であることから出力4桁目は”2”となる。以上より出力は”20f8”となる。

3つ目の出力である 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 型で出力する。PS4 コントローラと出力ポート Analog との関係を Table 4 に示す。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[2]~data[5]に関してのみ、後述するコンフィギュレーションパラメータで出力範囲の変更が可能である。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。

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
十字キー↓			↓キー押下時：1 出力は-1,0,1の3種
十字キー←	data[1]	0	←キー押下時：-1
十字キー→			→キー押下時：1 出力は-1,0,1の3種
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



範囲の変更可

Table 5 Configuration Parameter List

名称	データ型	デフォルト値	概要
m_Port	string	/dev/input/js0	接続デバイス先
m_Stick_Gain	double	1.0	スティックアナログ値ゲイン

本コンポーネントのコンフィギュレーションパラメータを Table 5 に示す。

m\_Port はコントローラの接続先アドレスのパラメータである。デフォルトの値は"/dev/input/js0"である。コントローラを PC につなげた際には、まずこの値を取るが、コントローラを 2 つ 3 つとつなげた場合などに"/dev/input/js1" , "/dev/input/js2"と変化する可能性がある。コンポーネントで扱いたいコントローラのアドレスを入力する。

m\_Stick\_Gain はスティックの取りうるアナログ値の範囲を変更するゲインであり、±m\_Stick\_Gain の範囲で出力をするように変更できる。たとえばデフォルト値 1.0 の場合は、-1.0~1.0 の範囲で出力するが、値を 12.5 に変更すると-12.5~12.5 の範囲で出力する。出力ポート Analog の data[2]~data[5]が該当する (Table 3)。他の出力 data[0], data[1],data[6], data[7]に関しては出力範囲の変更はされない。

## 改版履歴

Ver	改定日	内容
0.0	2016/3/30	新規作成