

# VR コンポーネント群解説マニュアル

名城大学メカトロニクス工学科  
ロボットシステムデザイン研究室

## 1. はじめに

### 1.1. 本書の目的

本書の目的は、VR 技術の既存システムへの簡易的な導入を目的として開発した RT コンポーネント群の仕様を解説するものである。

### 1.2. 概要

本書で開設する RT コンポーネント群には以下が含まれる

コンポーネント名	機能
ViveController	VR デバイスの一種である HTC VIVE の機能を簡易的に利用するための RT コンポーネントである。本コンポーネントは以下の 4 つの機能をもつ。 <ul style="list-style-type: none"><li>● VIVE コントローラの位置姿勢・速度/ボタン情報の管理</li><li>● VIVE HMD の位置姿勢・速度情報の管理</li><li>● VIVE トラッカーの位置姿勢・速度情報の管理</li><li>● VIVE HMD への画像の出力</li></ul>
SampleRobotController	ViveController が取得した情報を PC 画面上に出力する RT コンポーネントである

## 2. 開発環境

### 2.1. 開発環境一覧

PC	ZBOX-EN1070-U
OS	Windows 10
ソフトウェア	CMake 3.13.0 / OpenRTM-aist-1.1.2-RELEASE
ハードウェア	HTC VIVE / HTC VIVE トラッカー 2018
開発言語	C/C++

### 2.2. HTC VIVE / HTC VIVE トラッカー 2018

VR デバイスの 1 つである HTC VIVE と HTC VIVE トラッカー 2018 の機能について以下に示す。HTC VIVE にはヘッドマウントディスプレイ(以下,HMD), コントローラ, ベースステーションが含まれている。

Table 1 HTC VIVE / HTC VIVE トラッカー 2018

VIVE HMD		ユーザーの頭の動きを反映して、ユーザーに画像を出力するデバイス
VIVE コントローラ		ユーザーの手の動きを反映するデバイス。トリガーやパッドの機能が備わっている
VIVE トラッカー		様々な場所に取り付け可能なトラッキング用のデバイス
VIVEベース ステーション		HMD・コントローラ・トラッカーの位置姿勢・速度を推定するデバイス

### 3. 開発環境の構築

本コンポーネントに必要な開発環境の構築手順を以下に示す。ただし、今回は 32bit 版のライブラリを使用することとする。

#### 3.1. OpenVR のインストール

HTC VIVE を用いた開発をおこなうためのライブラリである OpenVR を以下のサイトからインストールする。

<https://github.com/ValveSoftware/openvr>

C ドライブ直下に置くものとし、環境変数に以下を追加する。

変数名	OPENVR_PATH
パス	C:\¥openvr-master

#### 3.2. GLEW のインストール

HTC HMD へ画像を出力するためコンピュータグラフィックスライブラリである OpenGL を使用する。今回は OpenGL の拡張機能を利用可能にするための補助ライブラリである GLEW を以下のサイトからインストールする。

<http://glew.sourceforge.net>

C ドライブ直下に置くものとし、環境変数に以下を追加する。

変数名	GLEW_PATH
パス	C:\¥glew-2.1.0

#### 3.3. GLFW のインストール

GLEW と共に、OpenGL の拡張機能を利用可能にするための補助ライブラリである GLFW を以下のサイトからインストールする。

<http://www.glfw.org/download.html>

C ドライブ直下に置くものとし、環境変数に以下を追加する。

変数名	GLFW_PATH
パス	C:\¥glfw-3.2.1.bin.WIN32

## 4. RT コンポーネントの仕様

### 4.1. ViveController

ViveController は, VR デバイスの一種である HTC VIVE の機能を簡易的に利用するための RT コンポーネントである. 本コンポーネントは Table 2~4 のインターフェースから構成されており, 以下の 4 つの機能をもつ.

- VIVE コントローラの位置姿勢・速度/ボタン情報の管理
- VIVE HMD の位置姿勢・速度情報の管理
- VIVE トラッカーの位置姿勢・速度情報の管理
- VIVE HMD への画像の出力

Table 2 VIVEController

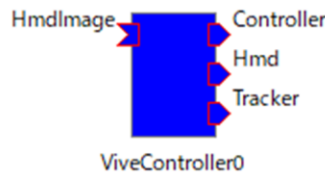
RTコンポーネントの名称		
ViveController		
データポート(入力)		
ポート名	データ型	説明
HmdImage	Img::TimedCameraImage	HMDに表示する画像データ
データポート(入力)		
ポート名	データ型	説明
Hmd	ViveControl::TimedVivePoseVelSeq	接続されている全ての VIVE HMDの情報
Controller	ViveControl::TimedViveControllerSeq	接続されている全ての VIVEコントローラの情報
Tracker	ViveControl::TimedVivePoseVelSeq	接続されている全ての VIVEトラッカーの情報

Table 3 VIVE HMD と VIVE トラッカーのデータ型

TimedVivePoseVelSeq		
接続されている全てのHMDまたはトラッカーの情報		
データ型	変数名	説明
RTC::Time	tm	時間
long	deviceNum	デバイスの接続数
Sequence<RTC::TimedPoseVel3D>	data	VIVEデバイスの位置姿勢・速度

Table 4 VIVE コントローラのデータ型

ViveController		
接続されている単一のVIVEコントローラの情報		
データ型	変数名	説明
RTC::PoseVel3D	controllerPoseVel	VIVEコントローラの位置姿勢・速度
boolean	gripButton	グリップボタン
boolean	applicationMenuButton	メニューボタン
boolean	systemButton	システムボタン
float	trigger	トリガーの引き具合 (0, 1)
float	padx	パッドx座標 (-1, 1)
float	pady	パッドy座標 (-1, 1)

Table 5 VIVE コントローラのデータ型

TimedViveControllerSeq		
接続されている全てのVIVEコントローラの情報		
データ型	変数名	説明
RTC::Time	tm	時間
long	deviceNum	デバイスの接続数
Sequence<ViveController>	data	接続されている単一のVIVEコントローラの情報

## 4.2. RT コンポーネントの利用方法

### 4.2.1. 使用方法 1: VIVE デバイスの情報の取得

本コンポーネントの使用方法的 1 つとして、VIVE デバイスの情報の取得が挙げられる。ここでは、SampleRobotControllerComp と組み合わせて PC 画面上に接続されている VIVE デバイスの位置情報を出力させる。実行した 2 つの RT コンポーネントを Fig. 1 のように接続し、コンポーネントをアクティベートする。VIVE HMD と VIVE コントローラを 1 つずつ接続したときの VIVE デバイスの情報取得結果を Fig. 2 に示す。

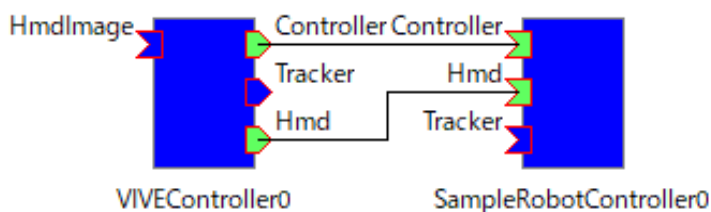


Fig. 1 接続図

```
C:\Users\rsdlab\workspace\SampleRobotController\build\src\Debug\SampleRobotControllerComp.exe
HMD0      posx:0.52  posy:0.79  posz:0.21
Controller0 posx:-0.46 posy:0.80  posz:0.17
```

Fig. 2 VIVE デバイスの情報取得結果

#### 4.2.2. 使用方法 2 : 画像データの出力

もう一つの使用方法として、HMD への画像データの出力が挙げられる．ここでは、既存コンポーネントである WebCameraComp と組み合わせて、HMD と PC 画面上に画像データを出力させる．WebCameraComp は以下のサイトからインストールする．

<https://github.com/rsdlab/WebCamera>

実行した 2 つの RT コンポーネントを Fig. 3 のように接続し、コンポーネントをアクティベートする．そのときの PC 画面上への画像データの出力結果を Fig. 4 に示す．

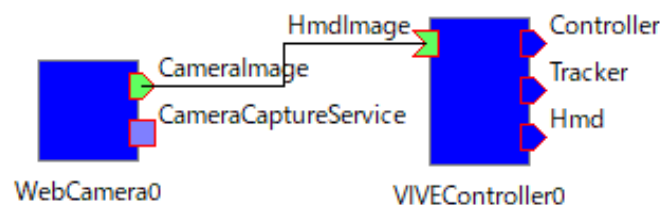


Fig. 3 接続図

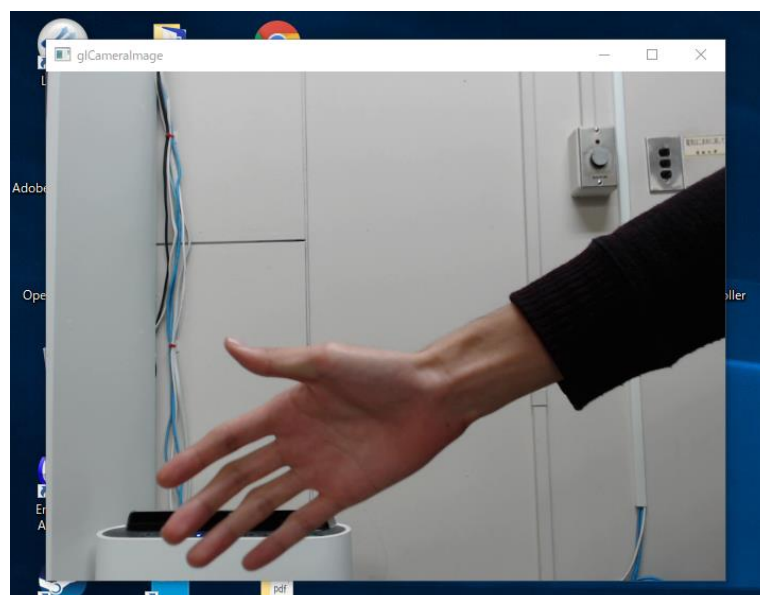


Fig. 4 画像データの出力結果

## 5. 今後の展望

本書にて，VR機能の利用を可能にする ViveController の使用方法を，サンプル RT コンポーネントである SampleRobotControllerComp と既存コンポーネントである WebCameraComp を用いて紹介した．今後は，本コンポーネントを既存のロボットの操作インターフェースとして利用したいと考えている．ViveController を様々なロボット操作に使用するために，VIVE の情報を利用したそれぞれのロボット用コントローラの開発を行ない，汎用性の向上を目指す．

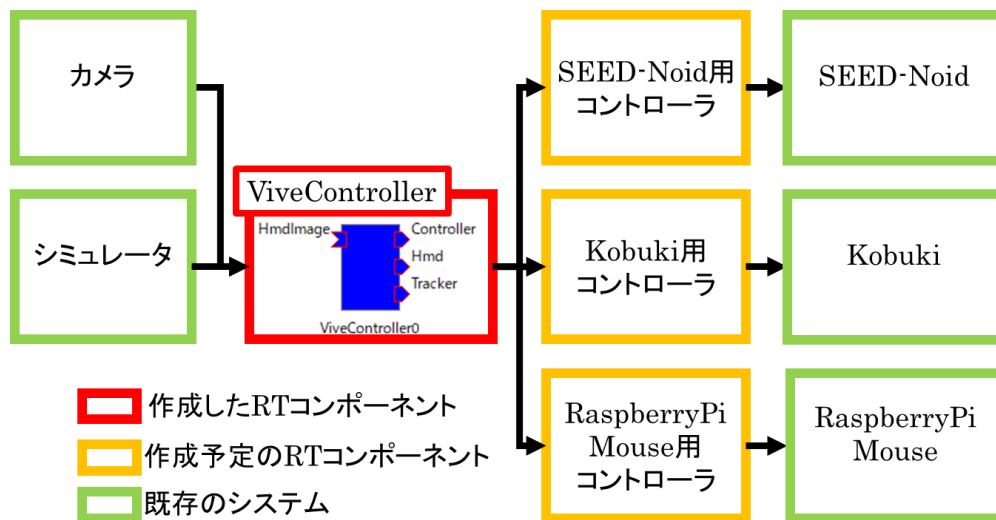


Fig. 5 VIVEController の利用例