

RT ミドルウェアコンテスト 2020

---

受付など通過歩行時の人数推定  
RTC の開発  
ユーザマニュアル

Ver.1.1

作成日 2020 年 11 月 6 日

芝浦工業大学 知能機械システム研究室

田島仁奈

# 目次

目次 .....	2
1. システムの概要 .....	3
1.1. はじめに .....	3
1.2. 開発したシステムの概要 .....	3
2. 開発・動作環境 .....	4
2.1. 使用機器 .....	4
2.2. ソフトウェア .....	4
3. システムの使用方法 .....	5
3.1. 事前準備 .....	5
3.2. システムの起動方法 .....	5
3.3. システムの終了方法 .....	10
4. 各 RTC 一覧と概要 .....	11
4.1. 開発した RTC 概要 .....	11
4.1.1. <i>ReceptionCommander</i> RTC .....	11
4.1.2. <i>logger</i> RTC .....	12
4.1.3. <i>gaussian_process_multi</i> RTC .....	13
4.2. 既存の RTC 概要 .....	13
4.2.1. <i>URG</i> RTC .....	13
4.2.2. <i>HumanTracking</i> RTC .....	14
5. 参考 .....	15
6. 連絡先 .....	15
7. 修正履歴 .....	15

# 1. システムの概要

## 1.1. はじめに

近年、日本では労働人口が減少傾向にあり、労働力が不足している。さらに働き方改革によって、今までより業務の効率化が求められてきている。業務効率化を図るために、ロボットを使用することが注目されている。ロボットには単純な業務を任せることで、人はその業務を行っていた時間をより高度な業務に使うことが出来る。単純な業務の1つとして、受付案内が挙げられる。受付案内の業務内容として、人数のカウントや混雑していない場所へ案内することが考えられる。それらのタスクを行うためには、人の軌跡の計測が必要である。これまで ApriPoco™（以降「Poco」と表記）を用いて受付応答の制御の研究が行われてきた。人の軌跡計測は Laser Range Finder（以降「LRF」と表記）を使用していたが、複数人数を測定するときに人が重なり LRF の死角にいる人が検出できず、人の軌跡が正確に測定されないことがあった。

そこで今回は複数人の軌跡を正確に測定するために、人の軌跡をガウス過程で予測する RTC を開発した。

## 1.2. 開発したシステムの概要

全体のシステムは、まず LRF から取得した値を処理し、人検出を行う。次に検出した人の座標の記録を行い、最後にガウス過程で人の軌跡予測を行うという流れである。人の座標の記録と人の軌跡の予測システムは司令塔から送られるコマンドから起動と停止を行う。今回開発したシステムは図 1 の赤枠で囲んだ ReceptionCommander RTC と logger RTC、gaussian\_process\_multi RTC である。既存の URG RTC と HumanTracking RTC に今回開発した RTC を組み込むことで人の軌跡予測を行う。今回開発した予測システムはリアルタイムで行うものではなく、人が検出範囲内に入ったときには座標記録を行い、人が検出範囲内から出たときにガウス過程で予測を行う。



Fig. 1-1 システムの流れ

## 2. 開発・動作環境

### 2.1. 使用機器

使用機器を以下の表 2-1 に示す。各々の使用は以下の通りである。

Table.2-1 使用機器一覧

ノート PC	OS : windows 10 , プロセッサ: Intel(R)Core(TM) i5-3340M CPU @2.70 GHz 実装メモリ: 8.0GHz
LRF	北陽電機株式会社 URG-04LX



Fig .2-1 LRF の外観

Table.2-2 LRF の仕様

測距範囲	距離	0.02~5.6m
	角度	240°
測距精度		0.06~1m: ±30mm、1~4m: 距離の3%
角度分解能		ステップ角: 約0.36° (360° / 1024分割)
走査時間		100ms/scan

### 2.2. ソフトウェア

- OpenRTM-aist1.2.0 32bit
- Python2.7.16 32bit
- Visual Studio Code

修正 BSD ライセンスとする。研究用途かつ利用者の責任下でご使用ください。

### 3. システムの使用方法

#### 3.1. 事前準備

以下の図のようにノート PC と LRF を接続し、電源ボタンを押す。



Fig. 3-1 LRF とノート PC の接続

#### 3.2. システムの起動方法

- ① 「OpenRTP x86」を起動する。

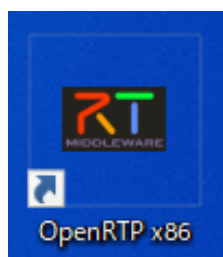


Fig. 3-2 OpenRTP x86 の画面上のアイコン

- ② workspace を選択し、起動をクリックする。

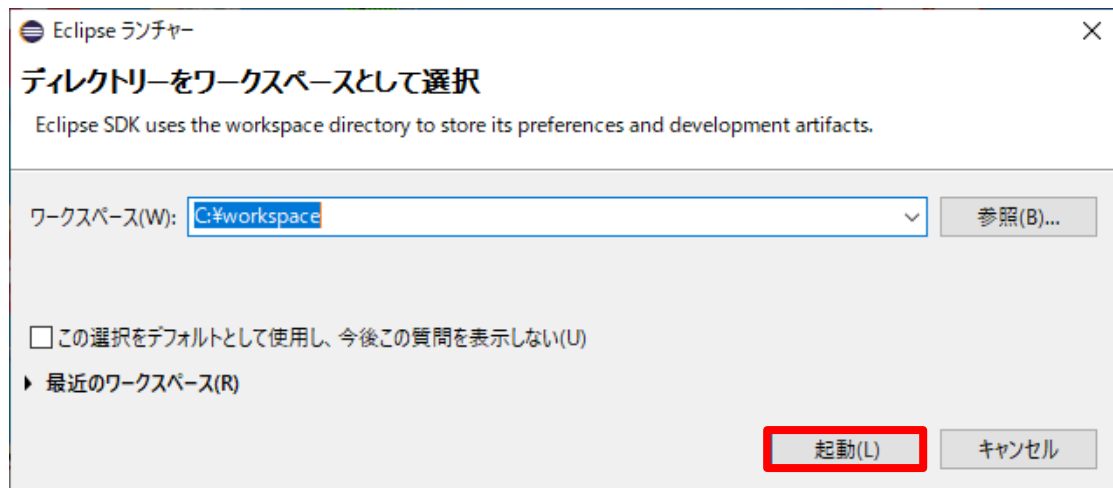


Fig. 3-3 workspace 選択画面

- ③ 図 3-4 の赤枠で囲った RT System Editor をクリックする。

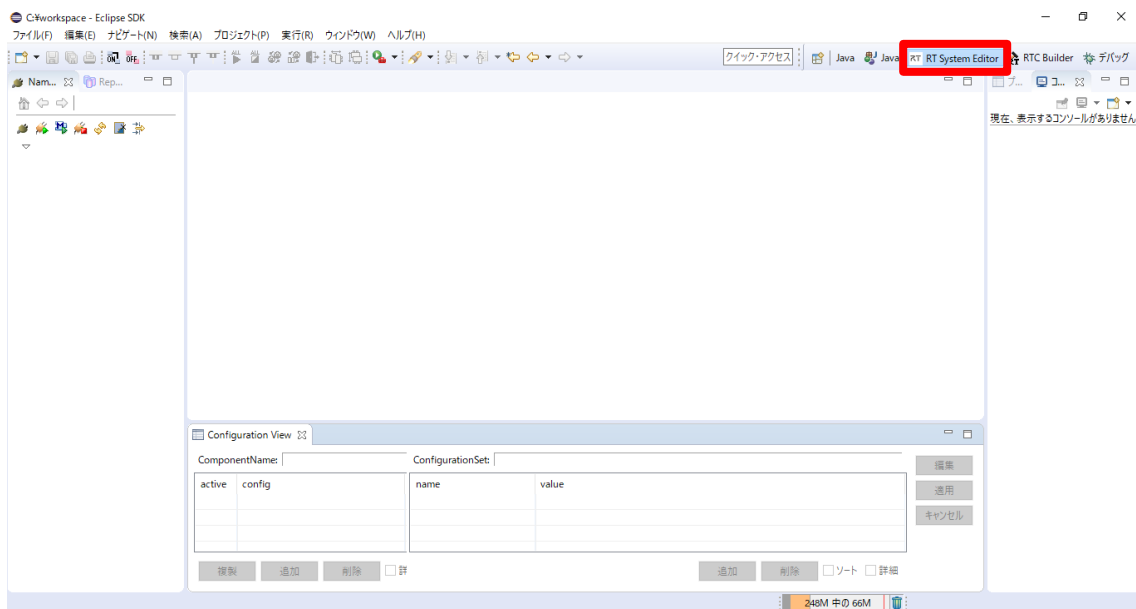


Fig. 3-4 RT System Editor 画面

- ④ 図 3-5 の赤枠をクリックすると、ネーミングサーバを起動する。ネーミングサーバが起動すると、図の下線が引いてあるような「localhost」が出現する。

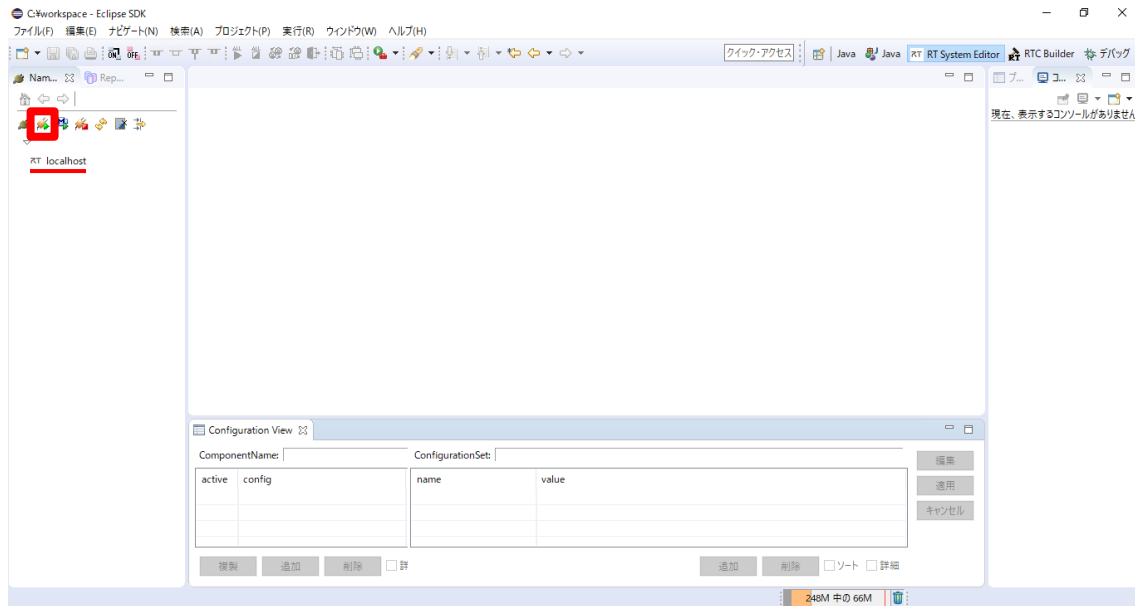


Fig. 3-5 ネーミングサーバ起動画面

- ⑤ 図 3-6 の赤枠をクリックすると、System Diagram が ON になる。

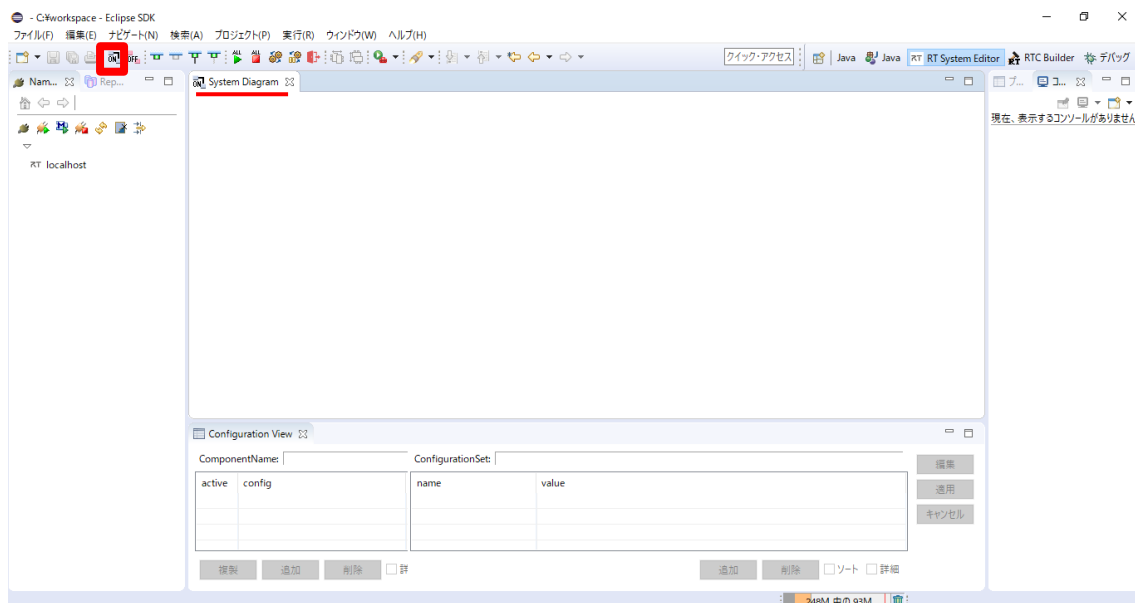


Fig. 3-6 System Diagram を ON にした画面

- ⑥ 「Auto\_connect\_Reception.bat」を起動する。起動すると、ネーミングサーバに図 3-8 のように表示される。

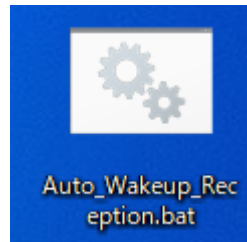


Fig. 3-7 Auto\_connect\_Reception.bat の画面上のアイコン

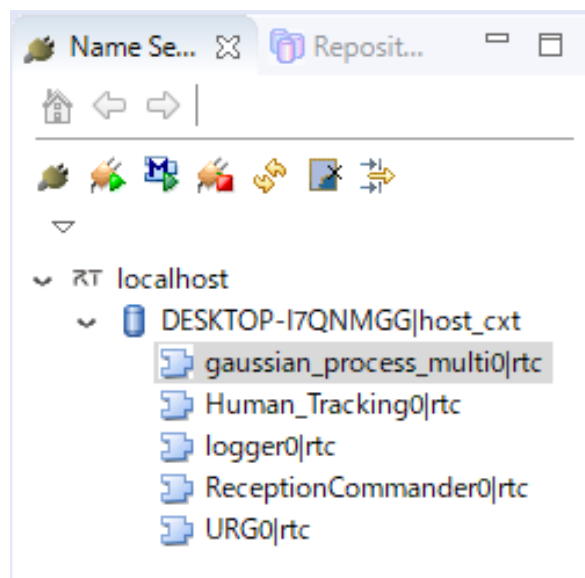


Fig. 3-8 ネーミングサーバ上に表示されたコンポーネント



- ⑦ ネーミングサーバから Open Diagram に RTC をドラッグし、ドロップする。図 3-9 のような状態になればよい。

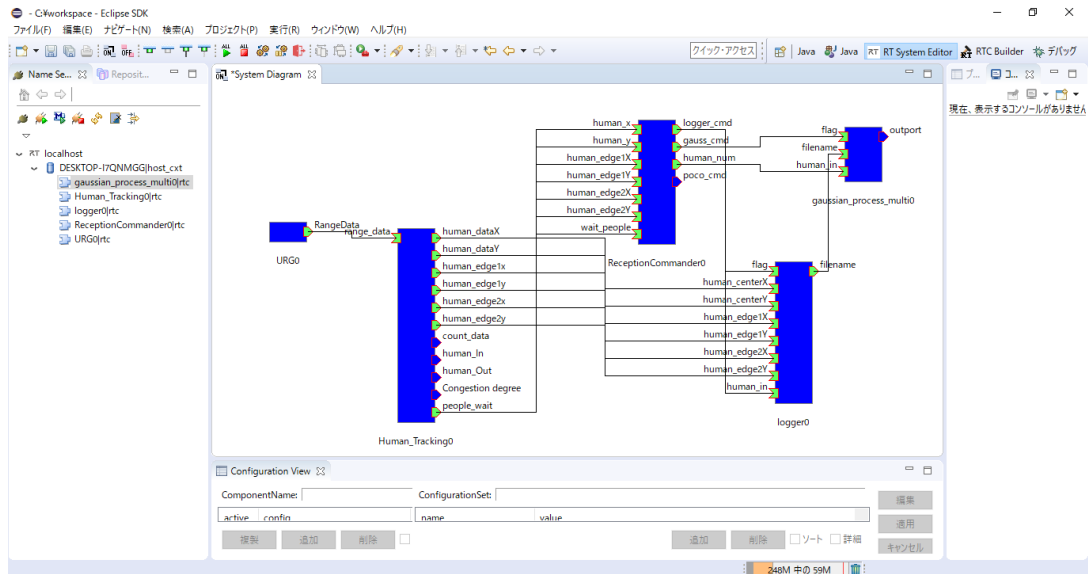


Fig. 3-9 Open Diagram にコンポーネントをドロップした画面

- ⑧ URG をクリックし、Configuration View を表示させる。図 3-10 で囲った赤枠の COM の値を変え、横にある「適用」をクリックする。デバイスマネージャから LRF の COM 番号を確認できる。

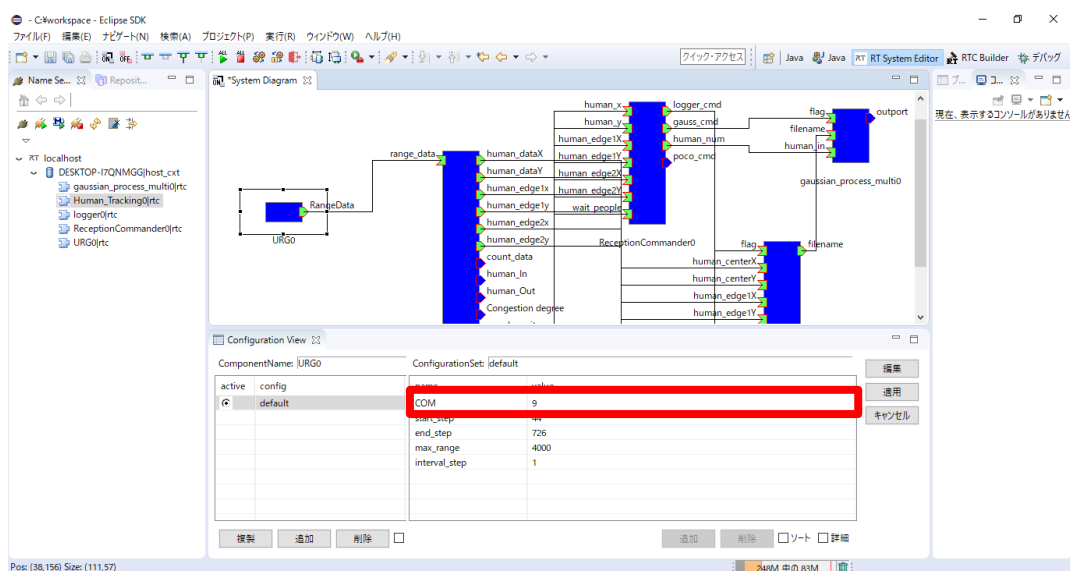


Fig. 3-10 Configuration View の COM 番号変更

⑨ 図 3-11 の赤枠をクリックし、「Activate Systems」をする。以上で起動は完了である。

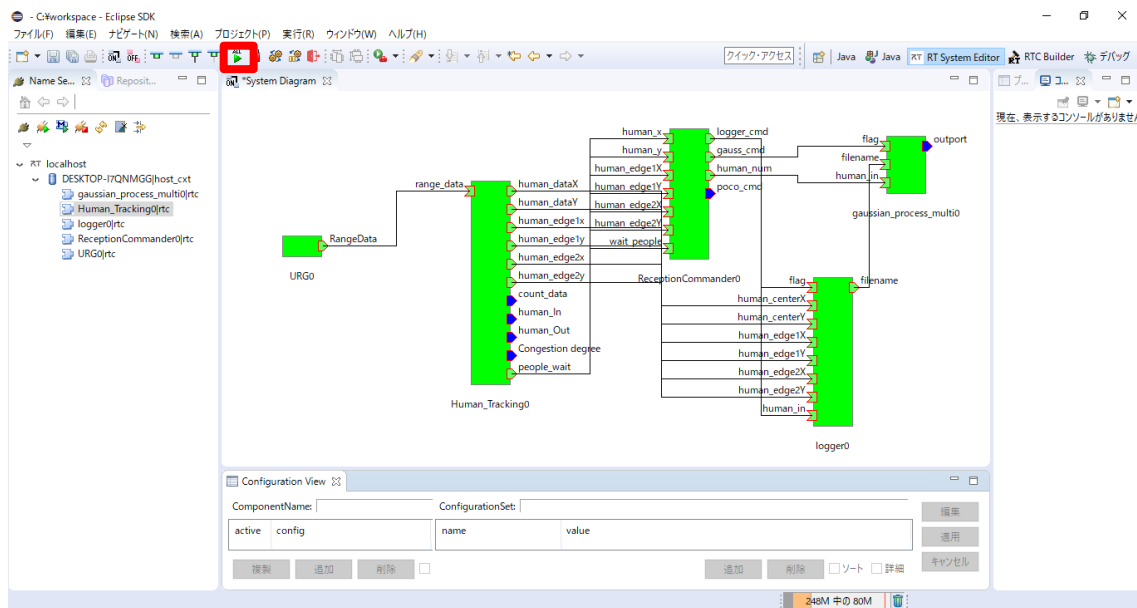


Fig. 3-11 「Activate Systems」した画面

### 3.3. システムの終了方法

図 3-12 の赤枠で囲った「Deactivate Systems」をクリックし、「kill\_Reception.bat」を起動させると、すべてのシステムが終了する。

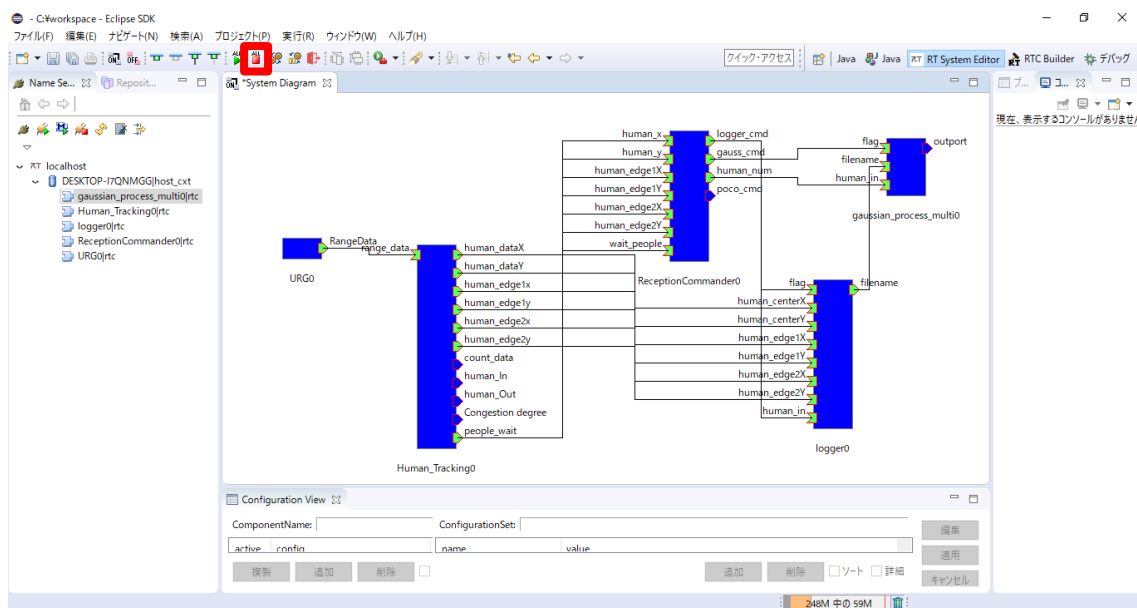


Fig. 3-12 「Deactivate Systems」をした画面

## 4. 各 RTC 一覧と概要

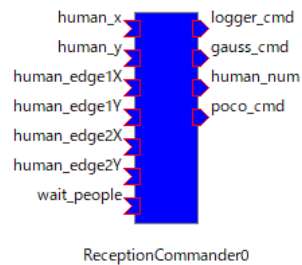
### 4.1. 開発した RTC 概要

#### 4.1.1. ReceptionCommander RTC

この RTC は HumanTracking RTC から人の中心座標を受け取り、その位置によって logger RTC と gaussian\_process\_multi RTC を起動させたり、停止させたりする。人が検出範囲内に侵入した際に logger RTC を起動させるコマンドを出力し、検出範囲内に人がいない際には gaussian\_process\_multi RTC を起動させるコマンドを出力する。

Table.4-1 ReceptionCommander RTC

### ReceptionCommander RTC

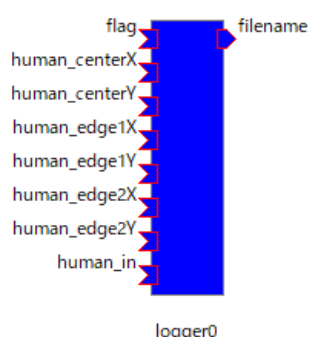


	名称	データ型	機能
InPort	human_x	TimedLongSeq	最大 10 人の人の中心座標 x を入力
	human_y	TimedLongSeq	最大 10 人の人の中心座標 y を入力
	human_edge1X	TimedLongSeq	最大 10 人の人の端点 x*を入力
	human_edge1Y	TimedLongSeq	最大 10 人の人の端点 y*を入力
	human_edge2X	TimedLongSeq	最大 10 人の人の端点 x*を入力
	human_edge2Y	TimedLongSeq	最大 10 人の人の端点 y*を入力
	wait_people	TimedLong	検出範囲内に残留している人の人数を入力
OutPort	logger_cmd	TimedString	logger RTC の起動と停止コマンドを出力
	gauss_cmd	TimedString	gaussian_process_multi RTC の起動と停止コマンドを出力
	human_num	TimedLong	特定の範囲内にいる人の人数を出力

4.1.2. logger RTC

この RTC は HumanTracking RTC から受け取った人の座標データを csv で保存する。  
ReceptionCommander RTC から起動コマンドが送られたときにだけ記録を行う。そのため、団体ごとの軌跡が csv に保存される。

Table.4-2 logger RTC

logger RTC			
			
	名称	データ型	機能
InPort	flag	TimedString	起動または停止のコマンドを入力
	human_centerX	TimedLongSeq	最大 10 人の人の中心座標 x を入力
	human_centerY	TimedLongSeq	最大 10 人の人の中心座標 y を入力
	human_edge1X	TimedLongSeq	最大 10 人の人の端点 x*を入力
	human_edge1Y	TimedLongSeq	最大 10 人の人の端点 y*を入力
	human_edge2X	TimedLongSeq	最大 10 人の人の端点 x*を入力
	human_edge2Y	TimedLongSeq	最大 10 人の人の端点 y*を入力
	human_in	TimedLong	検出範囲内の特定位置の人の人数を入力
OutPort	filename	TimedString	人の軌跡を保存したファイル名を出力

### 4.1.3. gaussian\_process\_multi RTC

この RTC は HumanTracking RTC から受け取った人の座標データをガウス過程で予測を行い、値を補正する RTC である。ReceptionCommander RTC から起動コマンドが送られる度に予測を行う。起動コマンドが送られると logger RTC から軌跡を記録した csv ファイル名が送られ、そのファイルを読み込み、予測を行った軌跡データが csv ファイルに保存される。

Table.4-3 gaussian\_process\_multi RTC



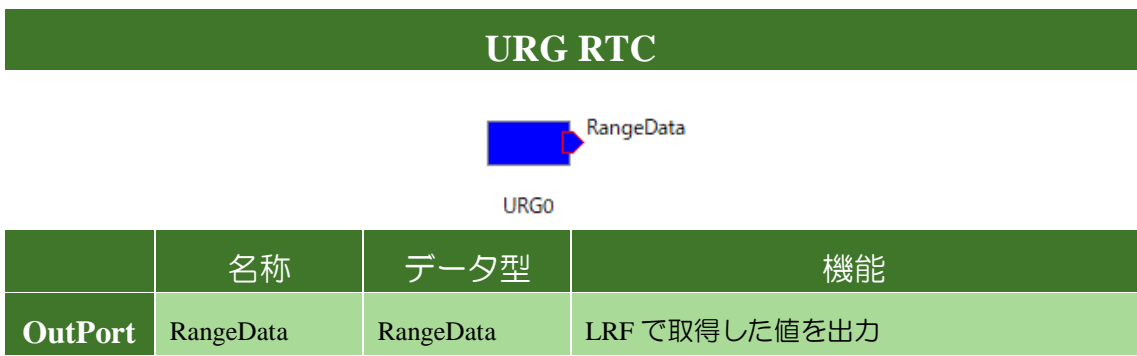
	名称	データ型	機能
<b>InPort</b>	flag	TimedString	起動または停止のコマンドを入力
	filename	TimedString	人の軌跡を保存したファイル名を入力
	human_in	TimedLong	検出範囲内の特定位置の人の人数を入力

## 4.2. 既存の RTC 概要

### 4.2.1. URG RTC

この RTC は LRF のセンサで測定した値を取得する。

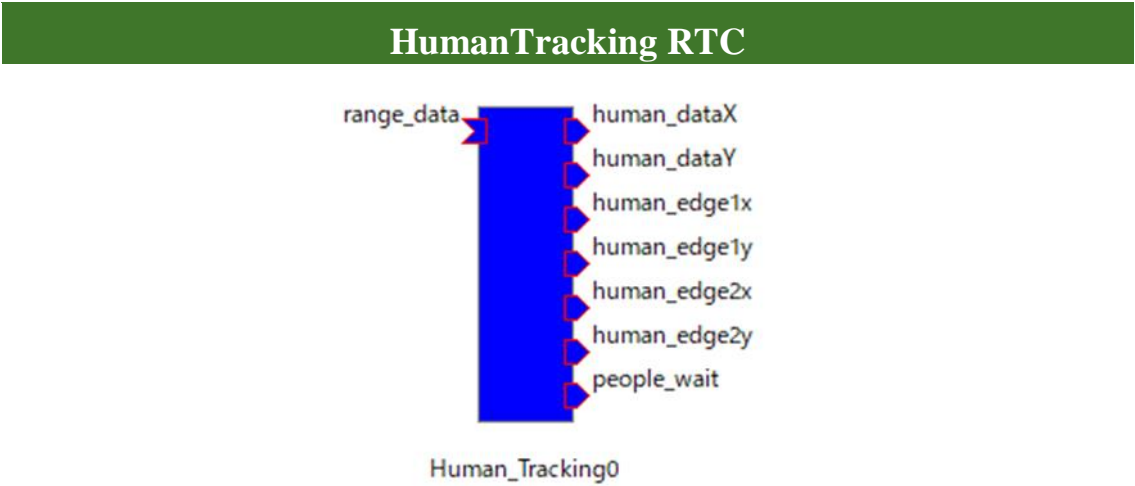
Table.4-4 URG RTC



4.2.2. HumanTracking RTC

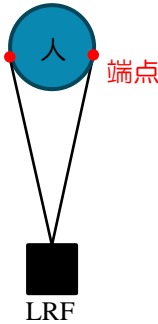
この RTC は人が検出範囲内存在する際に人の中心座標と端点，また残留している人の人数をカウントする。

Table.4-5 HumanTracking RTC



	名称	データ型	機能
InPort	range_data	RangeData	LRF から取得した値を入力
OutPort	human_dataX	TimedLongSeq	最大 10 人の人の中心座標 x を入力
	human_dataY	TimedLongSeq	最大 10 人の人の中心座標 y を入力
	human_edge1x	TimedLongSeq	最大 10 人の人の端点 x*を入力
	human_edge1y	TimedLongSeq	最大 10 人の人の端点 y*を入力
	human_edge2x	TimedLongSeq	最大 10 人の人の端点 x*を入力
	human_edge2y	TimedLongSeq	最大 10 人の人の端点 y*を入力
	People_wait	TimedLong	検出範囲内に残留している人の人数を入力

※人の端点：人にレーザが当たる端の座標



## 5. 参考

- [1] 厚生労働省：“企業規模別等でみた雇用人員判断 D.I.の推移”，  
(ア) <https://www.mhlw.go.jp/stf/wp/hakusyo/roudou/19/backdata/2-1-01.html>（閲覧日  
2020/10/19）
- [2] 岡野憲，松日楽 信人，太田麻美，加藤由花：“人の進行経路の分岐を予測 する RTC”，計測自動制御学会 SI 部門講演会, 2E1-02(2018)
- [3] 加藤由花，池田貴政，岡野憲，松日楽信人：インタフェースロボット応答制御のための歩行者分岐方向の予測，情報処理学会論文誌, Vol.60, No.2, pp.572-580(2019)
- [4] 畑尾直孝，鮫島一平，加賀美聡：角度ベース複数仮説を用いた LRF による複数種類・複数個の移動体追跡手法，計測自動制御学会論文集, Vol.51, No.5, pp.297-308(2015)
- [5] C. E. Rasmussen and C. K. I. Williams: Gaussian Processes for Machine Learning, The MIT Press (2006)

## 6. 連絡先

芝浦工業大学工学部機械機能工学科知能機械システム研究室

〒135-8548 東京都江東区豊洲3-7-5

E-mail : [md20053@shibaura-it.ac.jp](mailto:md20053@shibaura-it.ac.jp)（田島仁奈）

[matsuhir@shibaura-it.ac.jp](mailto:matsuhir@shibaura-it.ac.jp)（松日楽信人）

## 7. 修正履歴

以下に修正履歴を示す。

日付	バージョン	変更内容
11/20	Ver.1.1	起動方法について追記しました
11/26	Ver.1.2	人の端点についての記述を追記しました