

人との共存を目指した遠隔操作 ロボットの音声発話 RTC 手引書

第 1 版

芝浦工業大学大学院 理工学研究科 機械工学専攻

知能機械システム研究室

作成者氏名：飯田 悠貴

作成日時：2020 年 11 月 25 日

最終更新日：2020 年 11 月 25 日

目次

1. 初めに	2
1.1 本 RTC 作成の目的	2
1.2 遠隔操作システムの全体像	3
2. コンポーネントについて	3
2.1 音声出力 RTC 概要	3
2.2 RTC 開発環境	4
2.3 使用するハードウェア	5
2.4 音声出力の流れ	5
2.4.1 ゲームパッドによる操作情報取得	5
2.4.2 速度情報の取得	6
2.4.3 ロボットへの速度指令値に基づいた音声発話	7
2.4.4 本コンポーネントの汎用性について	7
3. 開発環境	9
4. ロボット遠隔操作システム	9
5. RTC のビルド手順	11
5.1 CMake を行い build フォルダの作成までの手順	11
5.2 Visual Studio による RTC のビルド, デバッグの手順	13
6. RT System Editor 上での動作確認	17
7. RTC 利用方法	20
7.1 移動ロボット用 PC の RTC 準備方法	20
7.2 操縦者用 PC の RTC 準備方法	23
付録 1. 開発環境 (Eclipse) の準備	25
1.1 Eclipse 概要	25
1.2 Java 実行環境 (JRE) のインストール	25
1.3 Eclipse のダウンロード	26
1.4 Eclipse の設定	28
付録 2. ネームサーバの準備手順	29
付録 3. CMake のインストール手順	34
参考文献	37

1. 初めに

1.1 本 RTC 作成の目的

新型コロナウイルスの感染が拡大し、全世界で感染者が続出している。感染者の看護をする際、フェイスシールドをしていても現場の看護師たちには常にウイルス感染のリスクが伴う。国際看護師協会（ICN）の情報によれば、世界で 9000 人以上の看護関連の職の方が新型コロナウイルスに感染してしまっている[1]。そのような危険な環境下では人の代わりに遠隔操作型ロボットを用いて行われている。感染が拡大するにつれて、遠隔操作ロボットの役割が明確になると言われており、需要が増大すると考えられる[2]。ロボットと人間が共存する環境では、ロボットと人間が安全に共存できなければならないが、ロボットの周囲の人間はロボットの動作状況を判断することが困難であり、接触事故に繋がる恐れがある。そこで、ロボットの動作状況を音声発話により周囲の人に知らせる RT コンポーネント（以下 RTC）を開発した。音声情報で人間とロボットの情報共有を目指し、安心・安全な共存環境の構築を目指す。

1.2 遠隔操作システムの全体像

開発したコンポーネントの使用環境例として、今回構築した移動ロボット遠隔操作システムを図1に示す。ロボットの遠隔操作にはRSNP(Robot Service Network Protocol)を用いた。操縦者はロボット上のカメラ映像を見ながらロボットをゲームパッドにより操縦する。

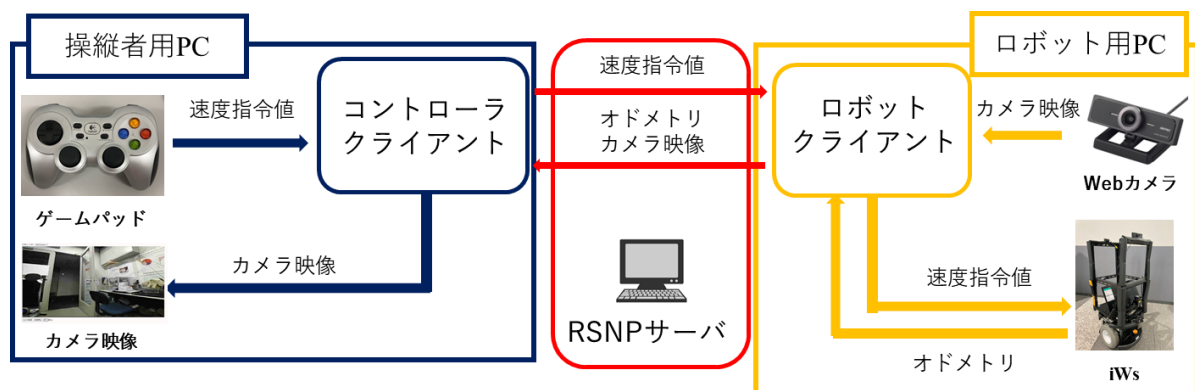


図1 遠隔操作システム概要

2. コンポーネントについて

2.1 音声出力 RTC 概要

今回作成した音声発話 RTC を表1に示す。今回使用するポートは Command_in であり、TimedVelocity2D 型の数値データを受け取る。ロボットの動作に対応した音声ファイル(wav形式)をそれぞれ用意し、受け取った数値データに対応した音声ファイルを読み込み、スピーカから発話する。発話内容とロボットの移動方向の関係を表2に示す。コンフィギュレーションパラメータとして、select_language と voice_interval を設定した。select_language は読み込む音声ファイルの言語を指定するパラメータであり、1番が日本語、2番が英語の音声ファイルに対応しており、それぞれの言語でロボットの状態を発話する。voice_interval はロボットの発話する間隔を変更できるパラメータであり、任意の整数値で定義することが可能となっている。

表 1 開発した RTC について

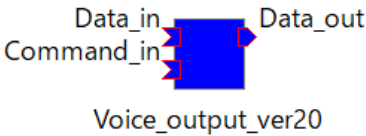
			
	名称	データ型	機能説明
InPort	Command_in	TimedVelocity2D	ロボット速度指令値の受信
	Data_in	TimedDoubleSeq	今回は未使用
OutPort	Data_out	TimedDoubleSeq	今回は未使用
Configuration	select_language	int	発話言語の変更用番号 1.日本語 2.英語
	voice_interval	int	音声発話の間隔変更

表 2 ロボットの移動方向と音声発話内容の関係

ロボットの移動方向	音声ファイル(発話内容)
右旋回	右旋回します
右折	右折します
前進	前進します
左折	左折します
左旋回	左旋回します
左方向後退	後退します
後退	
右方向後退	
停止時	停止中です

2.2 RTC 開発環境

筆者の開発環境を以下に示す.

- Windows 10 64bit
- OpenRTM-aist 1.1.2
- Java SE Development Kit 7u79 32bit
- Visual Studio 2015 (VC14)

- Phyton 2.7.9
- CMake-3.5.2
- Eclipse SDK Version: Luna SR2 (4.4.2)

ライセンスは MIT ライセンスとする．研究用とかつ利用者の責任で使用してください．

2.3 使用するハードウェア

今回使用する機器を以下に示す．

- 株式会社イクシス社製 iWs09 移動ロボット（対向 2 輪型移動ロボット）

（商品リンク：<https://www.ixs.co.jp/product/183>）

- Logicool Gamepad F310

（商品リンク：<https://gaming.logicool.co.jp/ja-jp/products/gamepads/f310-gamepad.html>）

- HP ProBook 650 G5（操縦者用 PC）

（商品リンク：https://jp.ext.hp.com/notebooks/business/probook_650_g5）

- 東芝 dynabook RZ63/A（ロボット側 PC）

（商品リンク：<https://dynabook.com/direct/pc/catalog/2016-fall-winter/rz63a/spec.html>）

2.4 音声出力の流れ

ここでは，開発した RTC が音声出力するまでについて説明する．

2.4.1 ゲームパッドによる操作情報取得

今回使用する GamePad_ver2RTC は DxLib が用いられており，ゲームパッドのジョイスティックのアナログ的操作量を得る．DxLib によりジョイスティックの操作量は左右・前後の入力状態を -1000 から 1000 までの範囲で表現された値を得ることが可能である．横方向入力が x 軸入力に対応し，上下方向の入力が y 軸方向入力に対応する．今回，スティックの座標は図 2 のように設定した．値はスティック操作量が初期状態から左側入力の時負の値となり，右側入力の時正の値となる．例えば，操作量が $(x, y) = (-1000, 0)$ の時，スティックは一番左，つまり y 軸負の方向にレバーが倒されていることを表し， $(x, y) = (0, 1000)$ の時は一番上，つまり y 軸正の方向にスティックが倒されていることを表す[3]．今回，図 3 のよう

にジョイスティックの入力角度 θ [deg]によって 8 つの領域に分割し, 移動指令をロボットに送信するように設定した.

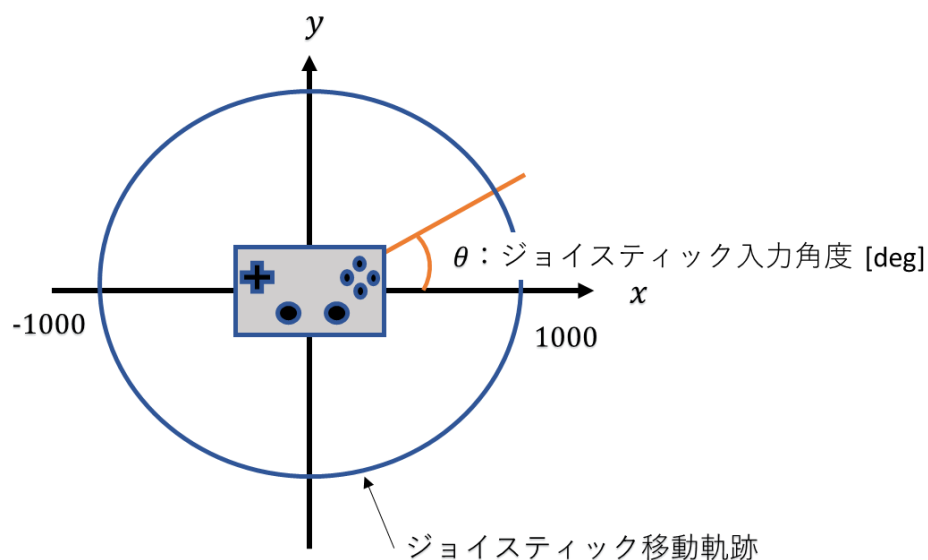


図2 ゲームパッドのジョイスティック座標設定

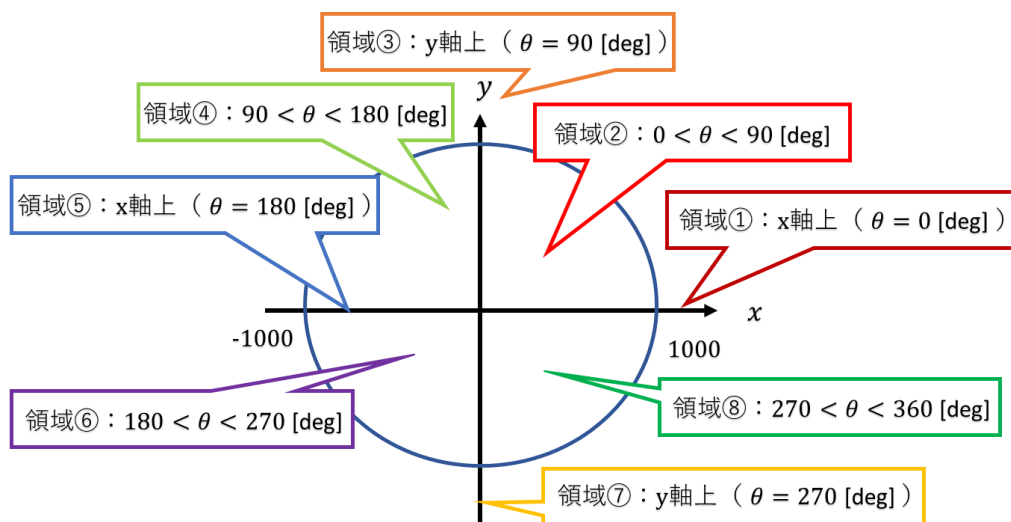


図3 ジョイスティック入力角度の分割

2.4.2 速度情報の取得

今回のシステムでは, スティックの入力量を二次元速度ベクトル (TimedVelocity2D 型)

で出力する。これにより、スティックの操作量が移動ロボットへの速度指令値となる。上記のデータ型で移動ロボットを操作する。図 4 に示すように、今回使用するデータは v_y m/s と v_a rad/s のみで、 $v_y = (y \text{ 方向入力値}) = (\text{前方速度指令値})$ 、 $v_a = (x \text{ 方向入力値}) = (\text{旋回速度指令値})$ に対応している[4]。速度情報は、移動ロボットのモータの回転方向を制御し、値の正負とモータ回転方向の正負は一致している。例えば、スティックを左に倒したとき、 $(v_y, v_a) = (0, -1000)$ となり、ロボットは左旋回運動を行う。ここまでの操縦者側 PC でのデータの流れであり、ControllerClientRTC に接続することでサーバにデータが送られる。



図 4 移動ロボットの速度指令設定

2.4.3 ロボットへの速度指令値に基づいた音声発話

サーバより送信されたデータは RobotClient_ver2 から移動ロボットの移動制御を行う iWs09RTC と音声発話を行う Voice_output_ver2RTC に TimedVelocity2D 型のまま送信される。音声発話 RTC 内では送信された (v_y, v_a) の値に基づく場合分けが行われており、速度指令値（スティック操作量）に対応した音声ファイルを出力する。今回の移動ロボットの移動方向とジョイスティックの入力量の関係を表 2 に示す。例えば、左にスティックを倒してロボットを操作する際、操縦者側 PC から $(v_y, v_a) = (0, -1000)$ の値がサーバ経由でロボット側 PC の音声発話 RTC に送られ、この (v_y, v_a) の正負、値の組み合わせに対応した音声ファイルを読み込み、ロボット側 PC のスピーカから「左旋回します」と発話を行う。

2.4.4 本コンポーネントの汎用性について

以上のように、開発した RTC はすべて操縦デバイスから出力された TimedVelocity2D 型を基準にして移動方向判断・場合分けを行っている。したがって、今回は表 3 のようにロボ

ットの移動方向を 8 領域で大別したが、例えば移動ロボットの「加減速」、「旋回の度合い」等さらに細かい動作を定義し、その動作に対応した音声発話を行うことが可能である。また、使用機器として対向 2 輪移動型ロボットを用いて遠隔操作システムを構築したが、次の条件を満たせば他の移動ロボット、操縦デバイスでも同様の効果が期待できると考えられる。

- ・ 操縦対象が移動指令を受信して移動制御を行うロボットであること
- ・ 操縦対象に移動指令を送信する RTC が TimedVelocity2D 型を出力すること
- ・ サーバを介して遠隔操縦者側とロボット側とでデータを授受するコンポーネントを使用すること

さらに、音声ファイルも wav 形式ならば任意の音声ファイルを作成し、発話を行うことも可能である。以上のように、ユーザの用途に応じて発話条件、使用ロボット、発話内容を変更可能な点で汎用性のあるコンポーネントだと考えている。

表 3 ジョイスティックの入力量とロボット移動方向の関係

領域	ジョイスティックの入力角度[deg]	移動方向
1	$\theta = 0$	右旋回
2	$0 < \theta < 90$	右折
3	$\theta = 90$	前進
4	$90 < \theta < 180$	左折
5	$\theta = 180$	左旋回
6	$180 < \theta < 270$	左方向後退
7	$\theta = 270$	後退
8	$270 < \theta < 360$	右方向後退

3. 開発環境

本 RTC を使用するには、以下のアプリケーションが必要となる。

- ・ Eclipse
- ・ CMake
- ・ Visual Studio
- ・ RTM ネームサーバ

付録でこれらのインストール方法を述べる。なお、インストールの手順はすべて Windows を使用して設定をしている。

4. ロボット遠隔操作システム

今回構築した移動ロボット遠隔操作システムを図 5 に示す。今回構築したシステムは次のような RTC で構成されている。

- ・ ゲームパッド等操作デバイスの操縦量を出力する RTC
- ・ サーバにデータを送信する RTC
- ・ サーバからデータを受信する RTC
- ・ 移動ロボットを制御する RTC
- ・ 音声発話を行う RTC

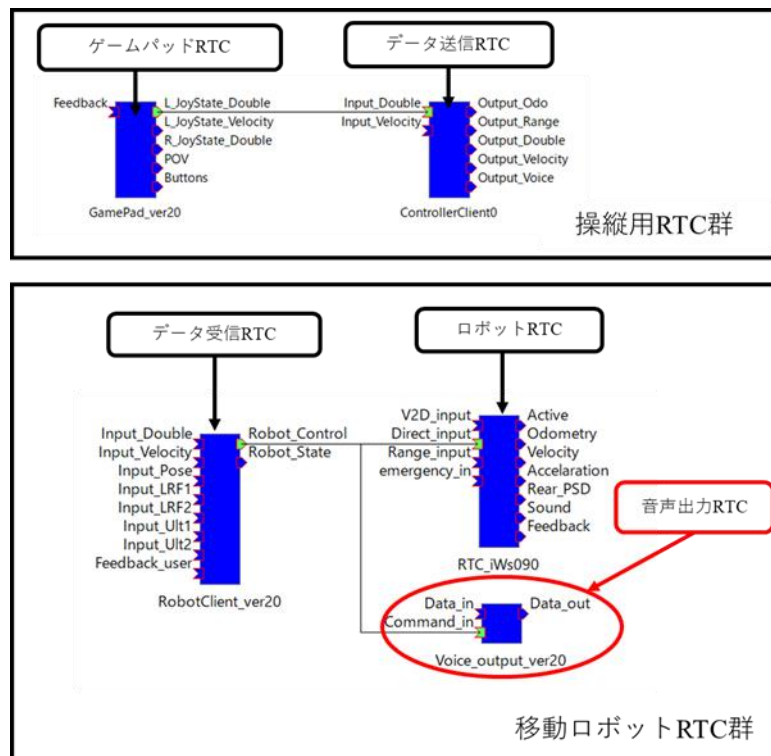


図5 移動ロボット遠隔操作システム全体図

5. RTC のビルド手順

ここでは開発した音声出力 RTC をビルドするまでの手順を述べる。

5.1 CMake を行い build フォルダの作成までの手順

1. CMake を起動する。PC 左下の「ここに入力して検索」欄に「CMake」と入力してアプリを起動する。CMake をインストールする際、デスクトップ上にショートカットを作成した場合、デスクトップ上のショートカットからでも起動することができる。
2. Voice_output_ver2 フォルダ内の「CMakeLists.txt」を CMake 上にドラック&ドロップする。
3. 「Where to build the binaries」の欄の「C:/Users/robo-lab/Desktop/Voice_output_ver2」最後に「/build」を書き加える。
4. Configure をクリックする。「Build directory does not exist, should I create it?」とでるので、Directory 以下を確認後、「Yes」をクリックする。

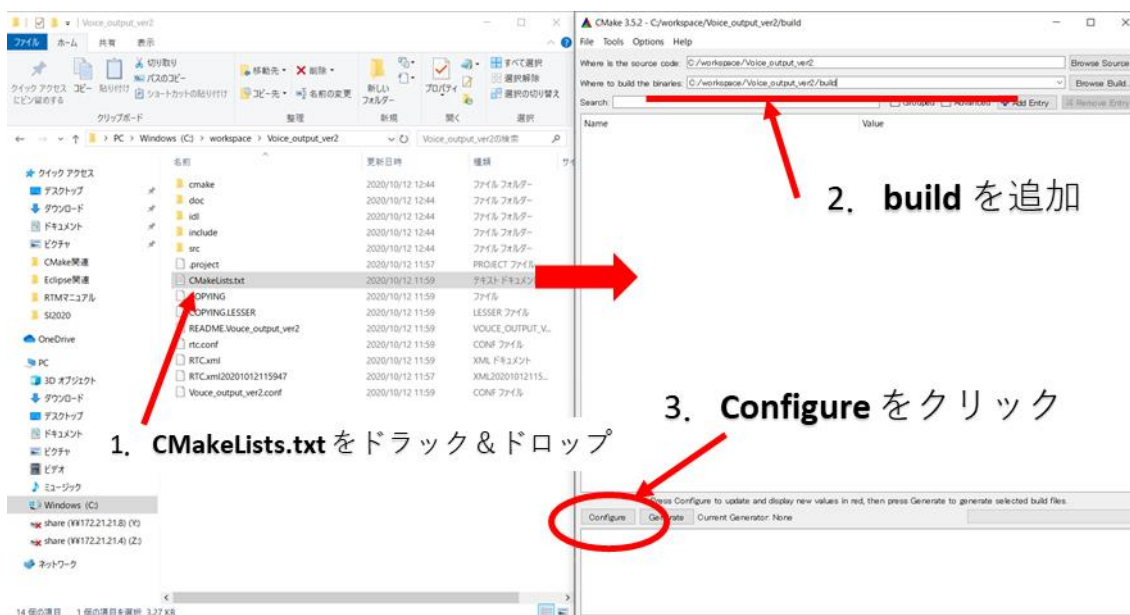


図6 build フォルダの作成①（手順1～3）

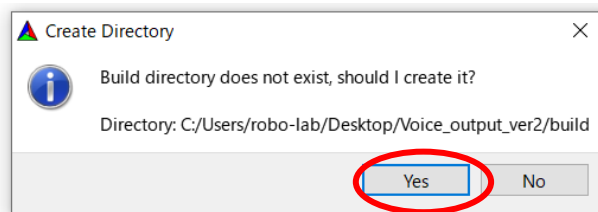


図7 build フォルダの作成②

5. クリック後図8のような画面になる。今回、Visual Studio 2015 の環境下でビルドを行うので、「Visual Studio 14 2015」で、「Use default native compilers」にチェックを入れて「Finish」をクリックする。

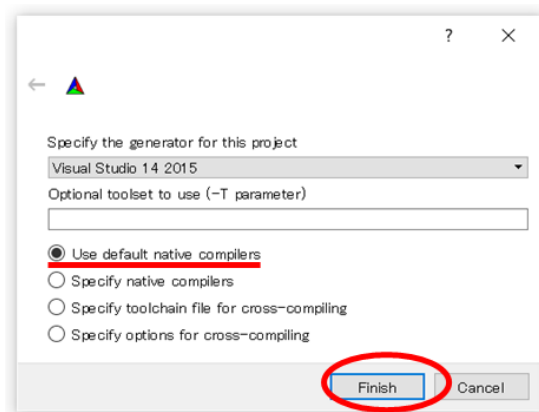


図8 コンパイラの設定

6. Configuring done と表示されているのを確認後、もう一度「Configure」をクリックする。その後、「Generate」ボタンをクリックし、ファイルを生成する。Generating done が表示されれば CMake 終了である。

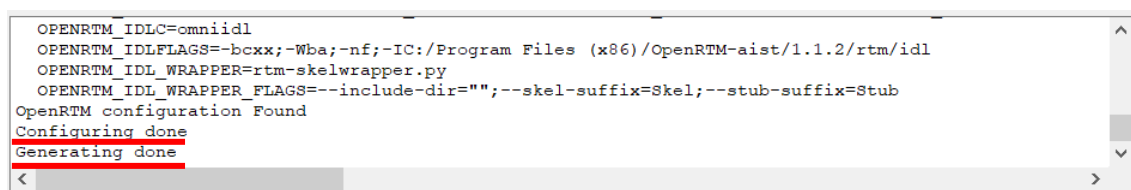


図9 CMake 終了時の画面

5.2 Visual Studio による RTC のビルド, デバッグの手順

1. Voice_output_ver2 フォルダ内に作成された build フォルダをクリックし、フォルダ内のソリューションファイル「.sln」を開く（ここで Visual Studio が起動する）。

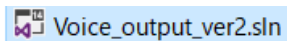


図 10 音声発話 RTC ソリューションファイル

2. Visual Studio の起動後、メニューバーの「ビルド」をクリックし、「ソリューションのビルド」をクリックしビルドを行う。



図 11 ビルド成功画面

3. ビルド成功後、音声ファイルの用意を行う。音声フォルダ内の音声ファイルを build 直下、および build/src/Debug にコピー&ペーストを行う。
4. 「ALL BUILD」を右クリックし、「プロパティ」をクリックする。プロパティページを開いた後、左側の欄の「構成プロパティ」より「デバッグ」を選択する。

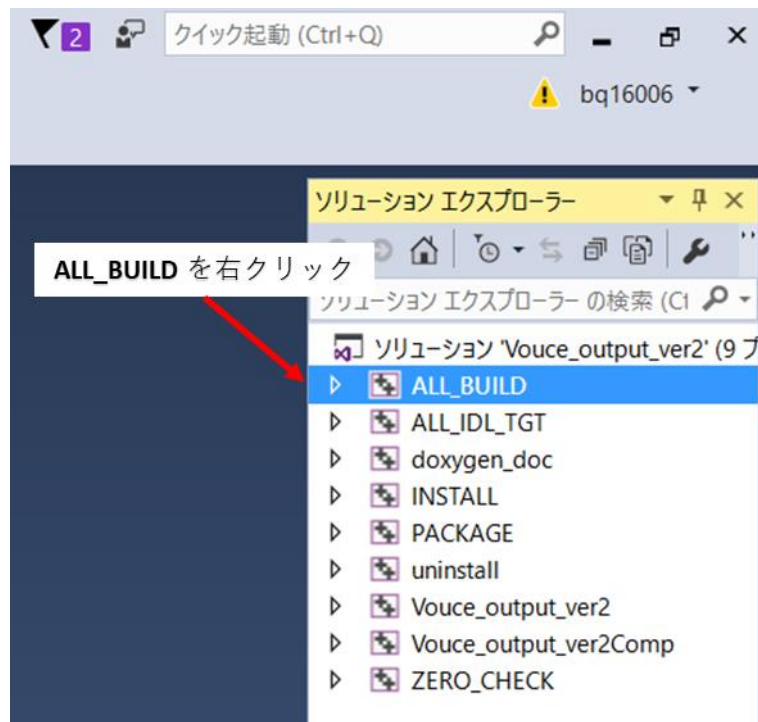


図 12 ALL_BUILD の選択

5. コマンド欄のボタンをクリックし、「参照」をクリックする。「build→src→Debug」の順にフォルダをクリックし、Debug フォルダ内の実行ファイル (.exe) をクリックする。
6. コマンド欄が図 15 のようにパスが通っていることを確認し、「適用」→「OK」の順にクリックする。

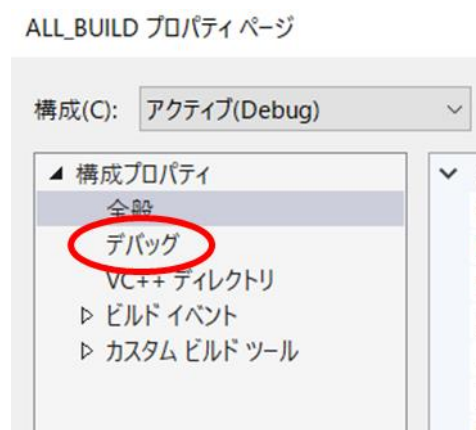


図 13 デバッグ選択画面

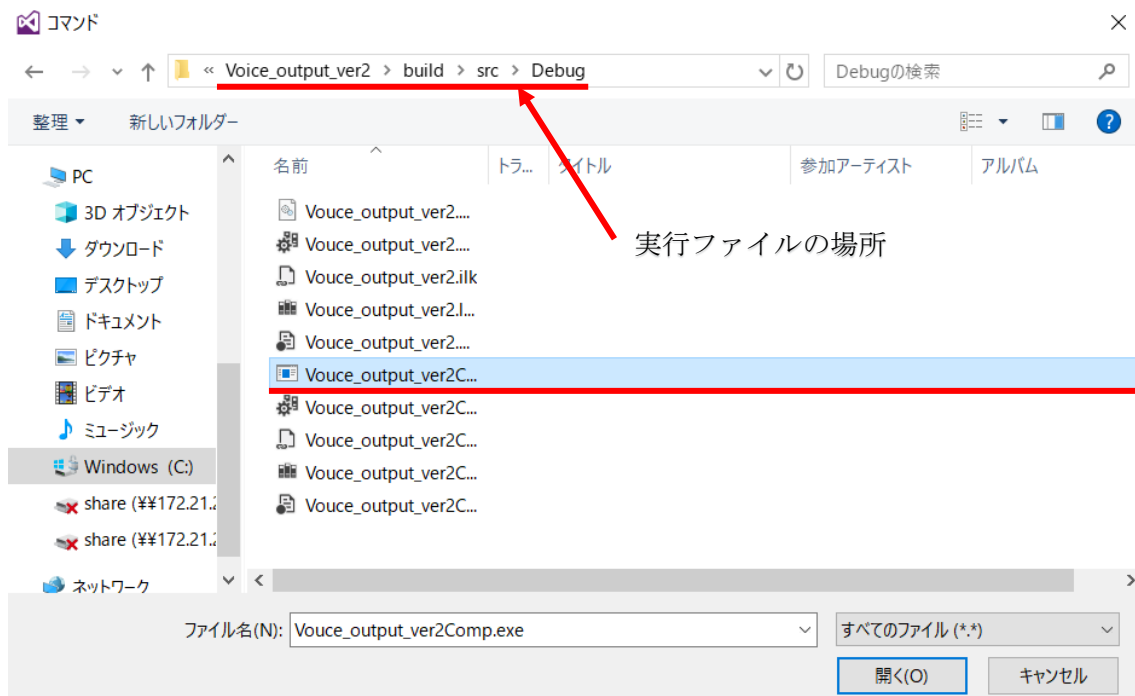


図 14 実行ファイルの選択

7. メニューバーの「デバッグ」から「デバッグの開始」をクリックする。デバッグに成功すると図 16 のようになる。

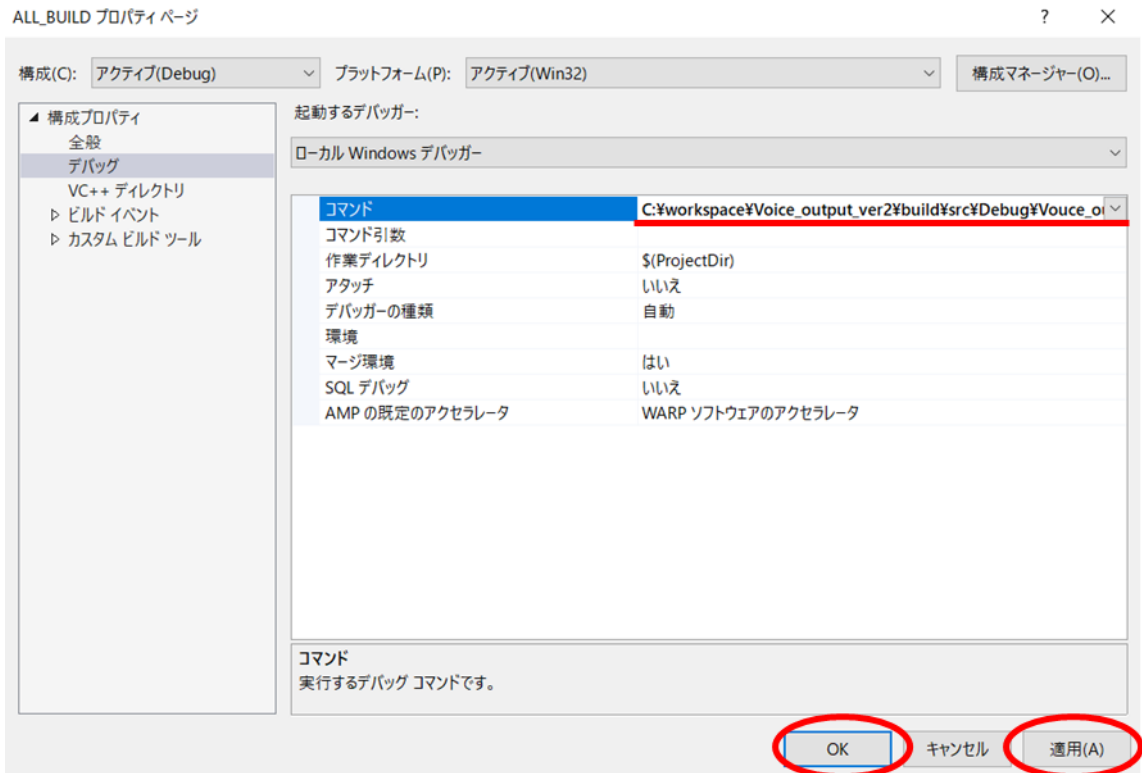


図 15 デバッグの設定

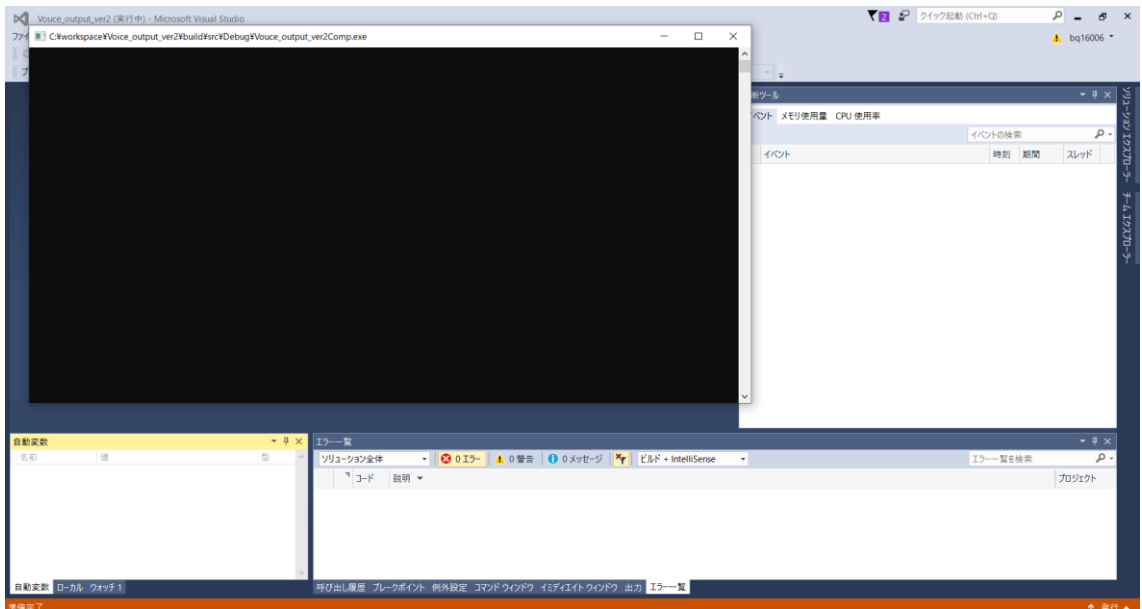


図 16 デバッグ完了画面

6. RT System Editor 上での動作確認

最後に、Eclipse 上の RT System Editor で動作確認を行う。

1. Eclipse を起動する。Eclipse のアイコンをクリック後ワークスペースを選択し、起動する。
2. 起動後、右上にある「パースペクティブを開く」をクリック後、「RT System Editor」を選択する。

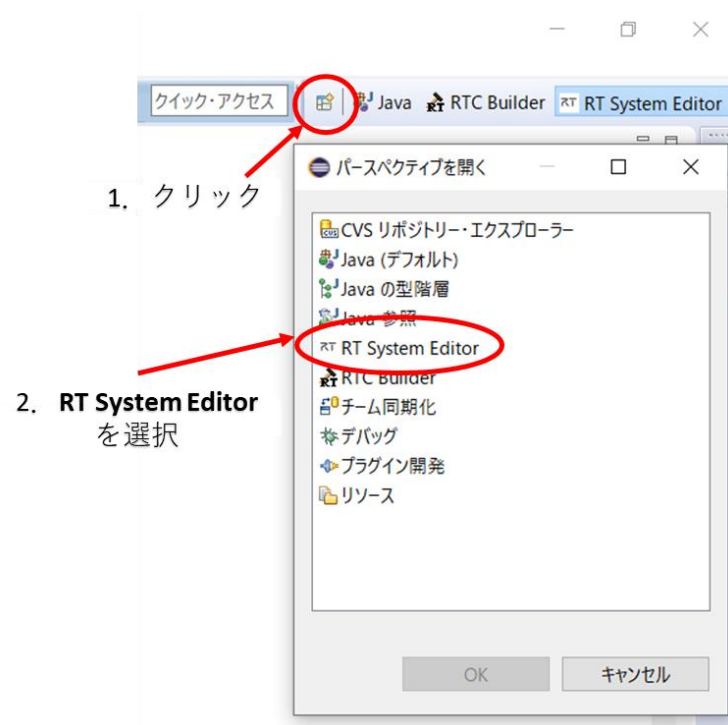


図 17 RT System Editor の開き方

3. ネームサーバを起動する。左下の「ここに入力して検索」欄に「Start Naming Service」と入力し、アプリを立ち上げる。
4. Eclipse に戻り、左側の▽ボタンよりネームサーバの追加をクリック、「ネームサーバのアドレスを入力してください。」となっているのを確認後「localhost」と入力する。

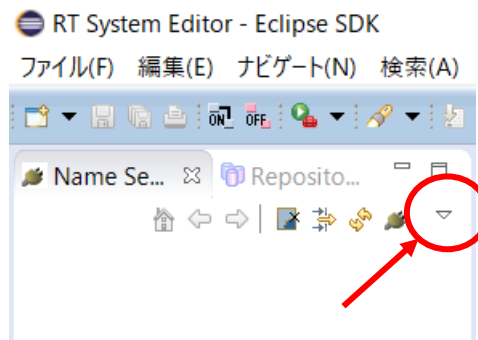


図 18 ネームサーバの設定

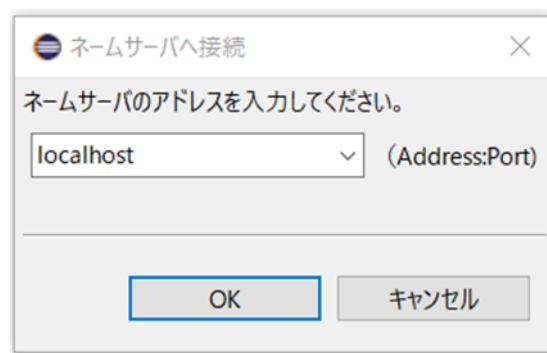


図 19 ネームサーバの接続

5. localhost と表示されているのを確認後, 図 20 のようにネームサービスビューに Voice_output_ver2 まで表示させる.
6. メニューバー左上の「ON」と書かれているボタンをクリックする. クリック後, 「System Diagram」が起動する.
7. ネームサービスビューの Voice_output_ver2 を System Diagram 上にドラック & ドロップすることで, コンポーネントが出現する.

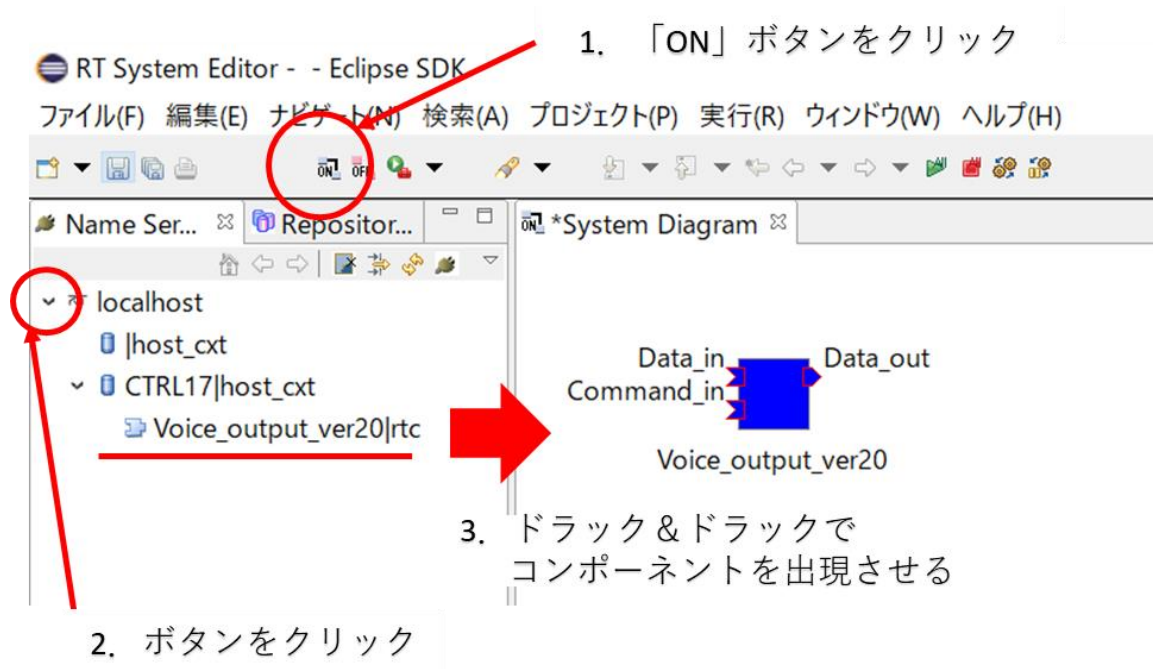


図 20 RTC 出現方法

8. 出現したコンポーネントを右クリックし、Activate をクリックする。図 21 のようにコンポーネントが緑色になれば動作確認完了である。

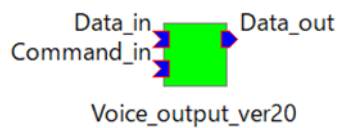


図 21 アクティベート済み RTC

7. RTC 利用方法

この章では開発した RTC を用いた移動ロボットの遠隔操作システムの利用方法について説明する。

7.1 移動ロボット用 PC の RTC 準備方法

1. 移動ロボットの電源を入れ、操縦者用 PC の Eclipse とネーミングサーバを起動する。
今回、ロボットの遠隔操作に RSNP 通信を用いるので、RSNP サーバを起動する。今回サーバも Eclipse を用いて構築されている。

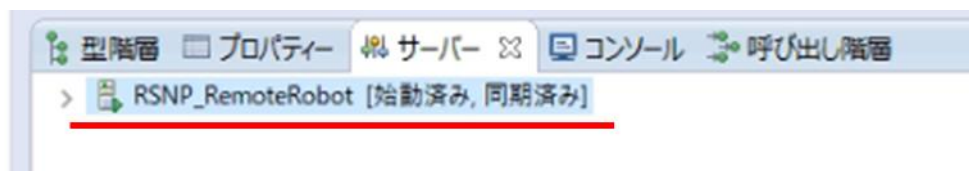


図 22 サーバ起動画面

2. RT System Editor 上でコンポーネント同士を接続する。図 23 のように「RobotClient_ver2」と「RTC_iWs09」, 「Voice_output_ver2」を接続する。接続時に「ConnectorProfile を入力してください」とメッセージが出るが、デフォルト状態のまま「OK」をクリックする。

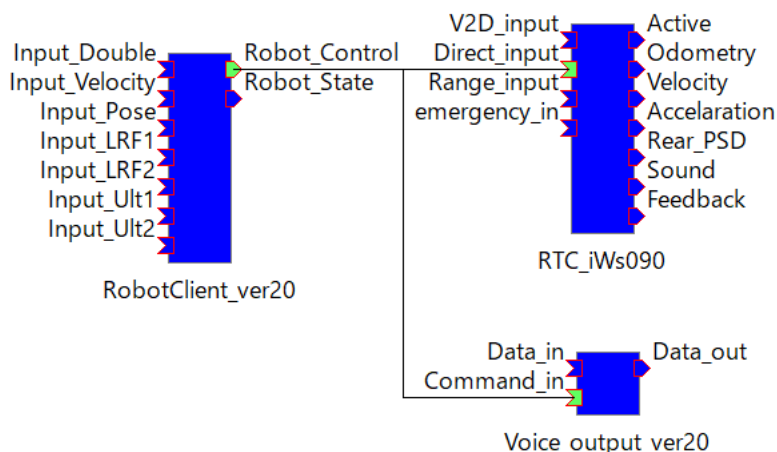
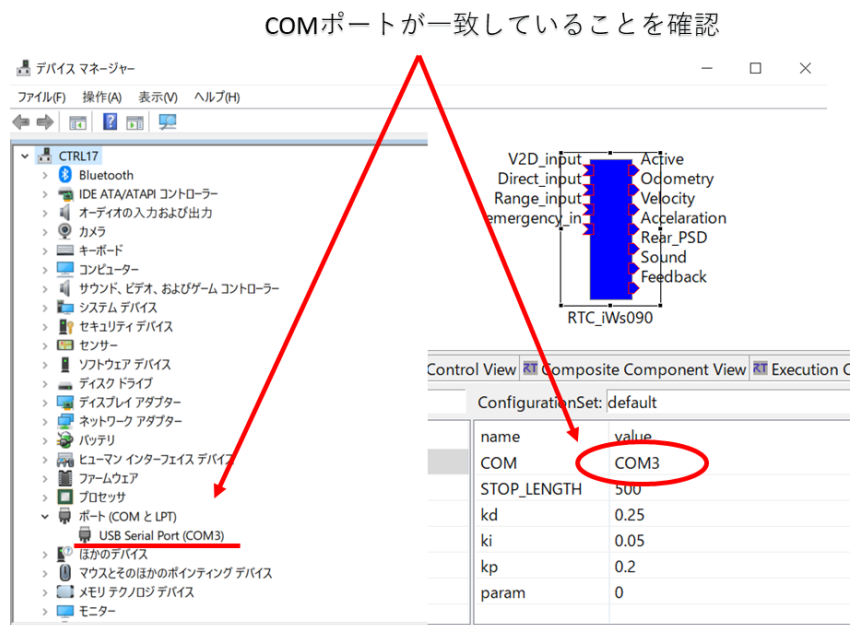


図 23 ロボット側 PC 遠隔操作システム

3. RTC_iWs09 はコンフィグレーションパラメータとして COM ポートを正確に設定しな

ければエラーが出るようになっている。デバイスマネージャを開き、移動ロボットの COM ポートを確認する (図 24)。



4. 各コンポーネントをアクティベートする。コンポーネントを右クリック、「Activate」をクリックし各コンポーネントをアクティベートする。一度に全てのコンポーネントを起動する際は、RT System Editor の「All Activate」ボタンをクリックする。図 26 のようにすべてのコンポーネントが緑色になったら、アクティベート完了である。



図 25 アクティベート選択画面



図 26 All Activate ボタン

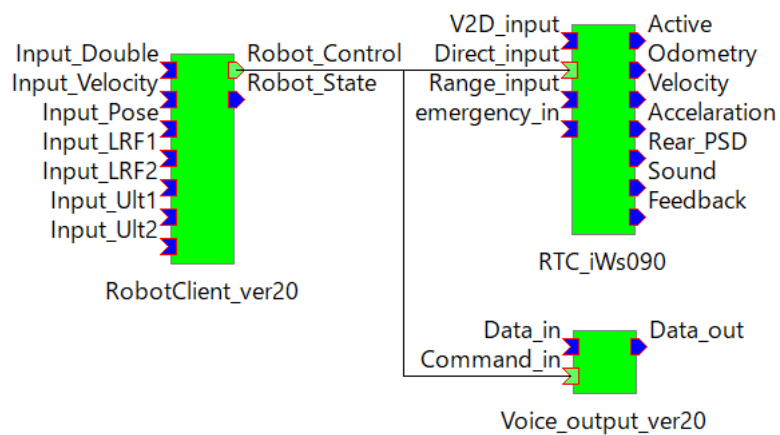


図 26 ロボット側コンポーネントアクティベート完了時の様子

7.2 操縦者用 PC の RTC 準備方法

1. ロボット側 PC のネーミングサーバ, Eclipse を起動する. 使用するゲームパッドを操縦者用 PC に接続させる. このとき, 操縦者側 PC のデバイスマネージャを開き, ゲームパッドが認識されているか (COM ポート番号が表示されているか) 確認する.
2. Eclipse の RT System Editorを開き, ロボット側 PC の設定時と同様にして「GamePad_ver2」 と 「ControllerClient」を接続させる.

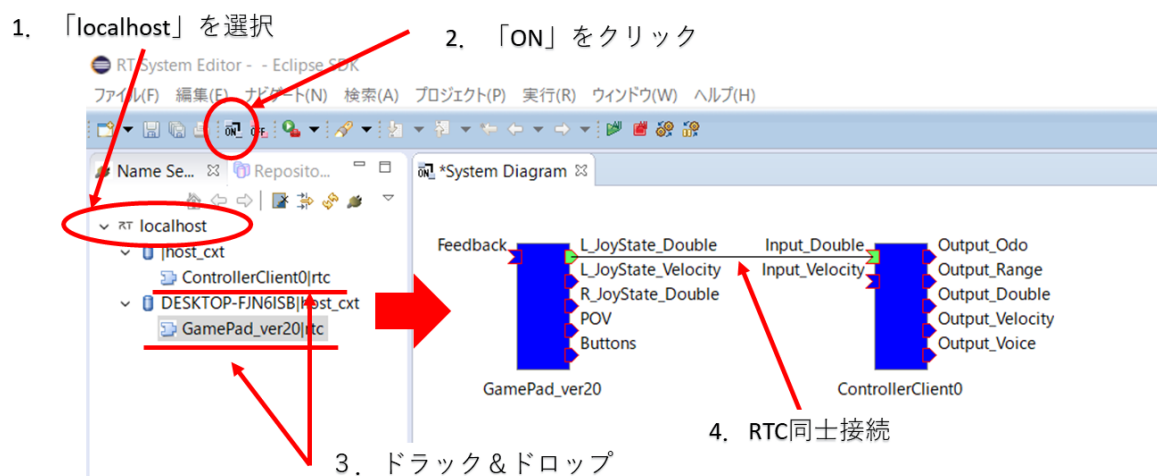


図 27 操縦者側 PC コンポーネント準備手順

3. 各コンポーネントをアクティベートする. 全てのコンポーネントが緑色になれば, アクティベート完了である.

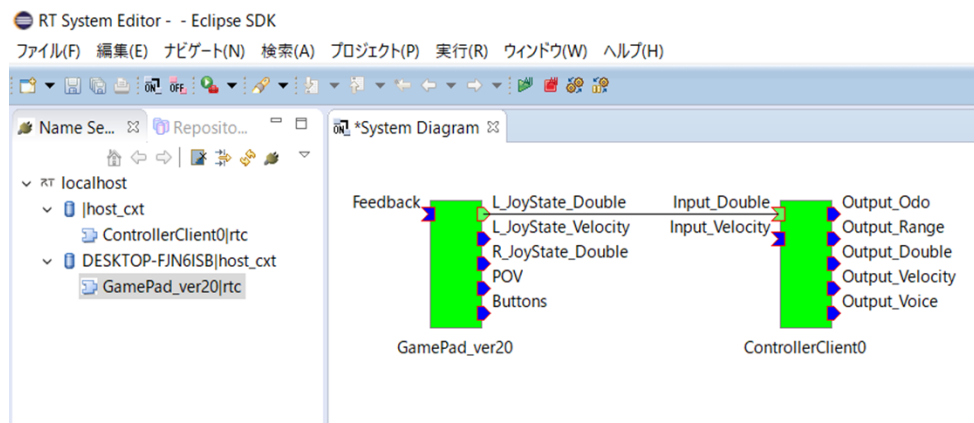


図 28 操縦者側コンポーネントアクティベート完了時の様子

付録 1. 開発環境 (Eclipse) の準備

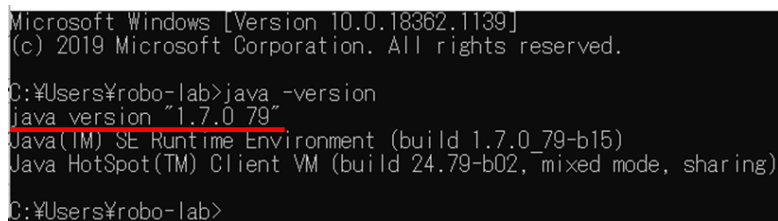
1.1 Eclipse 概要

Eclipse は Eclipse Foundation が開発するオープンソースの Java 言語, C++言語等様々な言語のための統合開発環境 (Integrated Development Environment: IDE) を構築するツールである。RT コンポーネントを使用する際に必要な RT System Editor を利用するには Eclipse をインストールする必要がある。

1.2 Java 実行環境 (JRE) のインストール

Eclipse は Java アプリケーションの一種であるため、Oracle の Java 実行環境 (JRE : Java Runtime Environment) をインストールする必要がある。既に JDK (Java Development Kits) 1.6 以上がインストールされている環境ではインストールする必要はない。Java 環境の確認方法を以下に示す。

1. PC 左下にある「ここに入力して検索」欄に「cmd」と入力し、コマンドプロンプトを開く。
2. コマンドプロンプトに「java -version」と入力する。
3. 一番上にある「java version」を確認する。



```
Microsoft Windows [Version 10.0.18362.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\robo-lab>java -version
java version "1.7.0_79"
Java(TM) SE Runtime Environment (build 1.7.0_79-b15)
Java HotSpot(TM) Client VM (build 24.79-b02, mixed mode, sharing)

C:\Users\robo-lab>
```

図 29 Java のバージョン確認

Java 本体のダウンロード手順は下記リンクを参考に行うこと。

(参考リンク : https://www.java.com/ja/download/help/windows_manual_download.html)

1.3 Eclipse のダウンロード

ここでは Eclipse 本体のダウンロード方法を説明する。

1. 下記 URL にアクセスし、「Pleiades All in One」をダウンロードする。最新版である Eclipse 2020 をクリックする。



図 30 Eclipse のバージョン選択

2. 下記ページの Windows 64bit の Full Edition 欄の Java のダウンロードボタンをクリックする。

MergeDoc Project

Pleiades 日本語化プラグイン

問題・不具合などの報告

簡単登録で無料ダウンロード

VAIO

特別付録「ノートPC購入に必要な
裏書きの作成について」もご好評
いただいております。

開く

Eclipse STS 入門ガイド

Spring Boot 入門

Jakarta EE Javadoc 日本語訳

JetBrains マニュアル日本語版

plugins、features ディレクトリーに格納されたプラグイン

dropins ディレクトリーに格納されたプラグイン

JDK などの各言語のコンパイラー、ランタイムなどの処理系

		Platform	Ultimate	Java	C/C++	PHP	Python
Windows 64bit 32bit は 2018-09 で終了	Full Edition	Download	Download	Download	Download	Download	Download
	Standard Edition	Download	Download	Download	Download	Download	Download
Mac 64bit Mac 版について (Qlite)	Full Edition	Download	Download	Download	Download	Download	Download
	Standard Edition	Download	Download	Download	Download	Download	Download
Eclipse 実行用 JDK 11		s	s	s	s	s	s
JDK 6u48 、 7u80 、 8u202 、 11.0.9 、 15.0.1		s	s	s	s	s	s
Lombok 1.18.16		s	s	s	s	s	s
Tomcat 6.0.53、 7.0.106 、 8.5.59 、 9.0.39		s	s	s	s	s	s
(Win のみ) MinGW 64bit 8.1.0		s	s	s	s	s	s
(Win のみ) Python 2.7.18 、 3.7.9		s	s	s	s	s	s
(Win のみ) XAMPP x64 (PHP) 7.4.11		s	s	s	s	s	s
Pleiades 2020.11.01		d	d	d	d	d	d
Autodetect Encoding 1.8.5		d	d	d	d	d	d
Platform 4.18.0		p	p	p	p	p	p
Buildship 3.1.4		p	p	p	p	p	p
CDT 10.0.0		d	d	d	d	d	d
EGit 5.9.0		p	p	p	p	p	p
GEE 3.11.0		p	p	p	p	p	p
JDT 3.18.300		p	p	p	p	p	p
m2e 1.5.3 (-wtp、-apt 含む)		p	p	p	p	p	p

図 31 Eclipse ダウンロード選択画面

- クリック後、リンクが出現するので、それをクリックし zip ファイルをダウンロードする。ダウンロードには約 16 分かかる。
- ダウンロード後、zip ファイルを解凍する。なお zip を解凍する際、「ファイルの名前を短くする」もしくは「7-Zip で解凍する」のどちらかを選択すること。7-Zip のダウンロードについては <https://sevenzip.osdn.jp/> を参照すること。
- ファイルの解凍後、Eclipse を起動する。ワークスペースは各自で設定すること。



図 32 Eclipse 起動画面

1.4 Eclipse の設定

1. Eclipse 起動後，ウィンドウタブをクリック後，設定をクリックする。
2. 右側にあるツリーメニューから「Java」→「インストール済みの JRE」の順に押し，「追加」ボタンを押す。
3. JRE の追加ウィンドウの一番下にある標準 VM を選択，「次へ」をクリックする。

ディレクトリーをクリック．インストール済み JDK のディレクトリーを指定する．JDK ディレクトリーの確認方法は下記リンクを参照すること．チェックボックスにチェックを入れれば設定完了となる．

(参考リンク：<http://www.maroon.dti.ne.jp/koten-kairo/works/Java3D/install0.html>)

付録 2. ネームサーバの準備手順

1. ネームサーバを利用するには、OpenRTM-aist 1.1.2 をインストールする必要がある。インストール時に Python2.7 が必要となるので、下記リンクよりインストールを行う。

(参考リンク：<https://www.python.org/downloads/release/python-270/>)

2. Python2.7 インストール後、下記リンクより OpenRTM-aist1.1.2 のインストールを行う。Windows インストーラ 32bit 版を選択しダウンロードする。

(参考リンク：<https://openrtm.org/openrtm/ja/node/6877>)

32bit用

Windows 用インストーラ (OpenRTM-aist、C++、 Python、Java版 および OpenRTP、rtshell(4.1.0) 含む) (Visual Studio 2008、2010、 2012、2013、2015 共通)	OpenRTM-aist-1.1.2- RELEASE_x86.msi MD5:59be8603f3fc007c2aed447f05 2886ce	2016/05/2 7
Python-2.7	python-2.7.10.msi	python.org
PyYAML	PyYAML-3.11.win32-py2.7.exe	pyyaml.org
CMake	cmake-3.5.2-win32-x86.msi	cmake
Doxygen	doxygen-1.8.11-setup.exe	doxygen

図 33 OpenRTM-aist インストーラのダウンロード画面

3. ダウンロード完了後、インストーラを起動してインストールを続ける。「次へ」をクリックする。



図 34 インストール画面

4. 同意書のページになる。チェックボックスにチェックを入れ、「次へ」をクリックする。

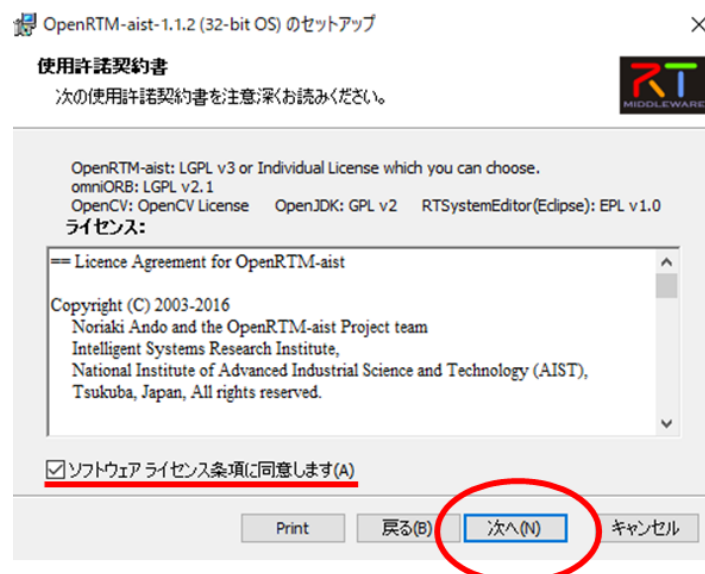


図 35 同意書画面

5. 次にインストーラ種類選択のページになる。「このコンピュータのすべてのユーザーにインストール」を選択し、「次へ」をクリックする。

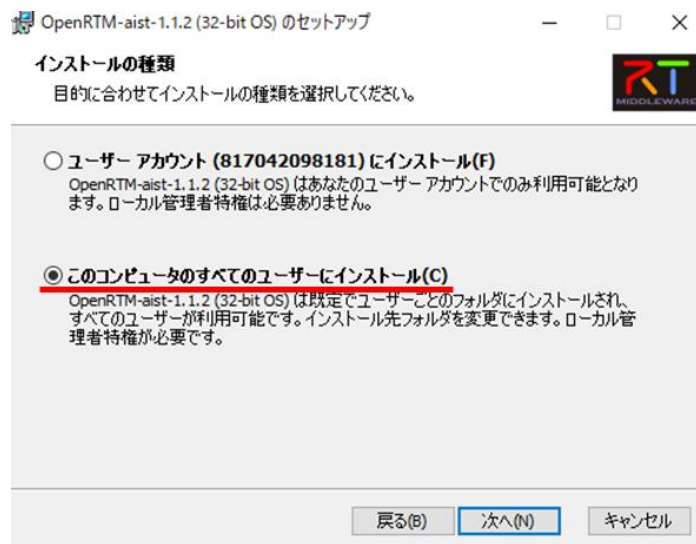


図 36 インストール画面

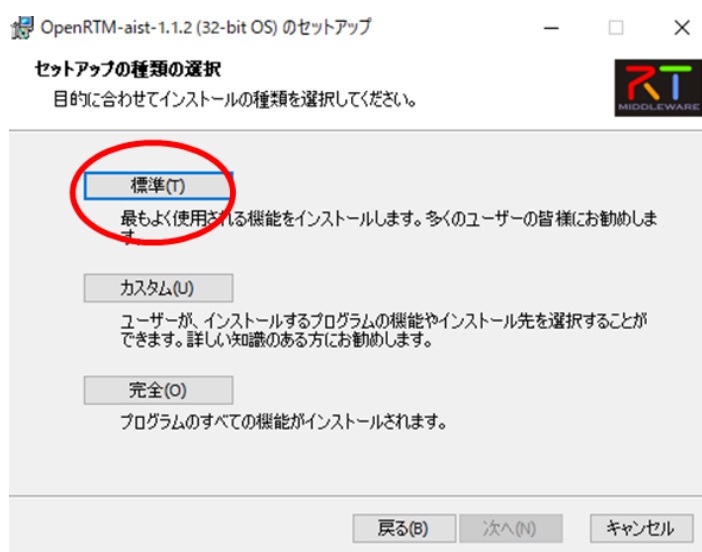


図 37 セットアップ画面

6. セットアップの種類を選択する。「標準」を選択し、「次へ」をクリックする



図 38 インストール準備完了画面

7. セットアップ完了後、インストール画面になる。「インストール」をクリックし、インストールを行う。
8. インストール完了後、「C:¥Program Files (x86)¥OpenRTM-aist¥1.1.2¥bin」にアクセスし、ネームサーバ (bat ファイル) を起動する。クリック後図 40 のような表示が出るが、「アクセスを許可する」をクリックすると起動する。

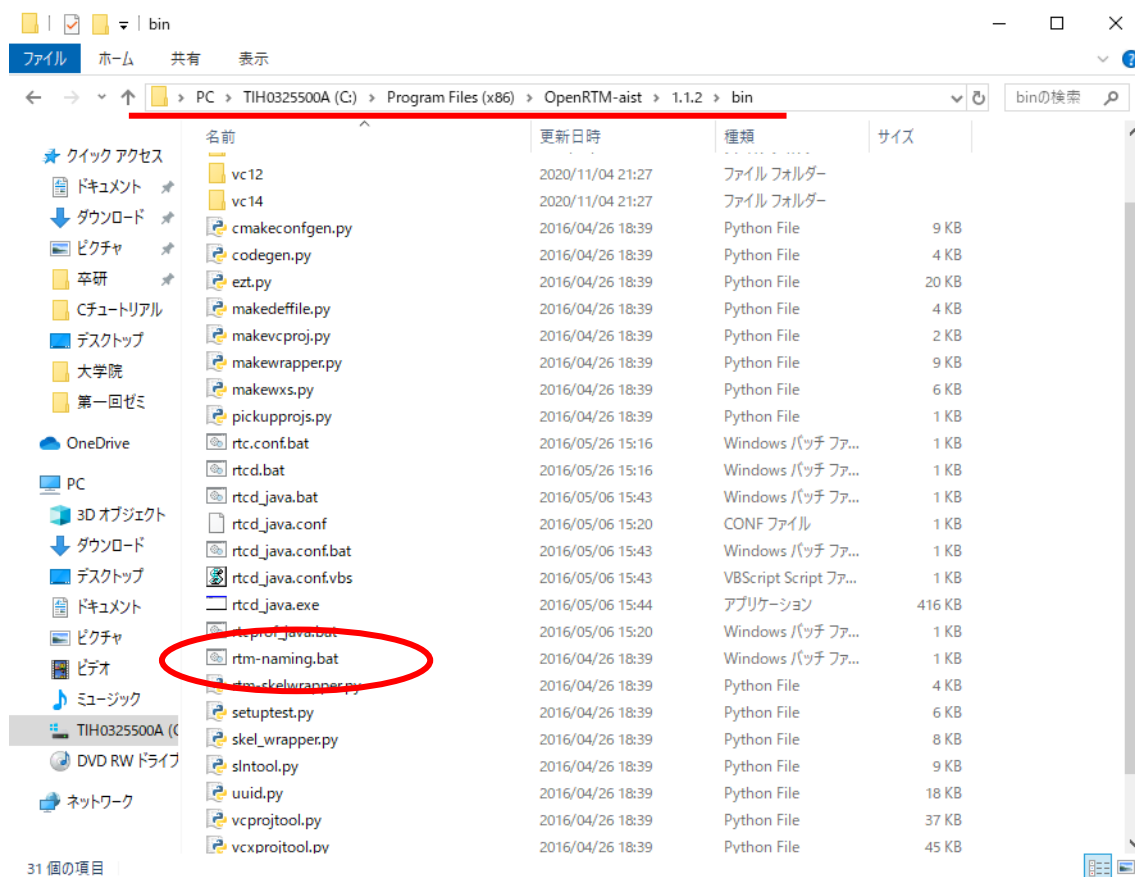


図 39 ネームサーバの場所

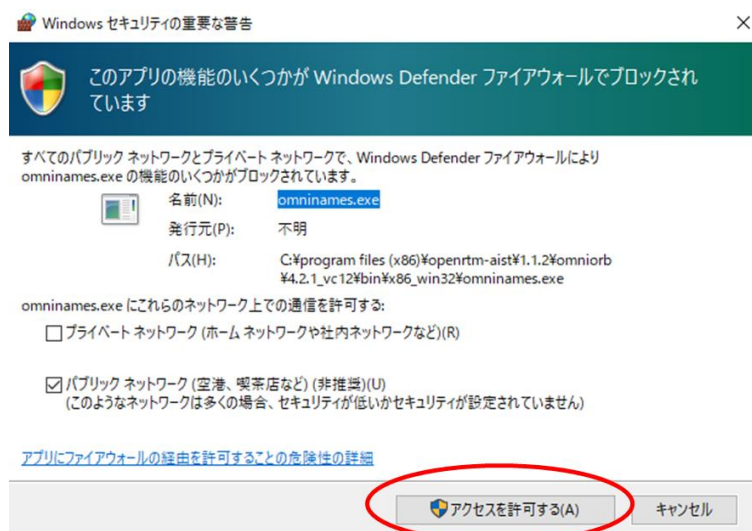


図 40 ネームサーバ立ち上げ時の表示

付録 3. CMake のインストール手順

ここでは開発した RTC をビルド可能にする「CMake」というアプリのインストール手順について述べる。なお、CMake3.4 以下のバージョンを既に持っている方は、一旦アンインストールしてから以下の手順で再インストールすること。

1. CMake をダウンロードする。下記リンクにアクセスし、Windows win64-x64 Installer 欄のリンクをクリックし、ダウンロードする。

(参考リンク：<https://cmake.org/download>)

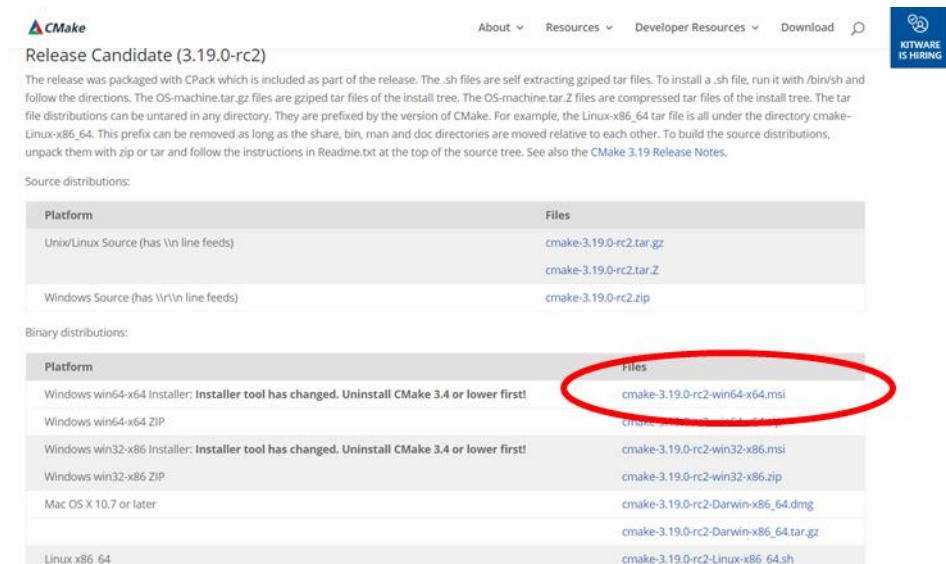


図 41 Windows インストーラ選択画面

2. ダウンロード後 CMake を起動する。「Next」をクリックし、インストールを開始する。

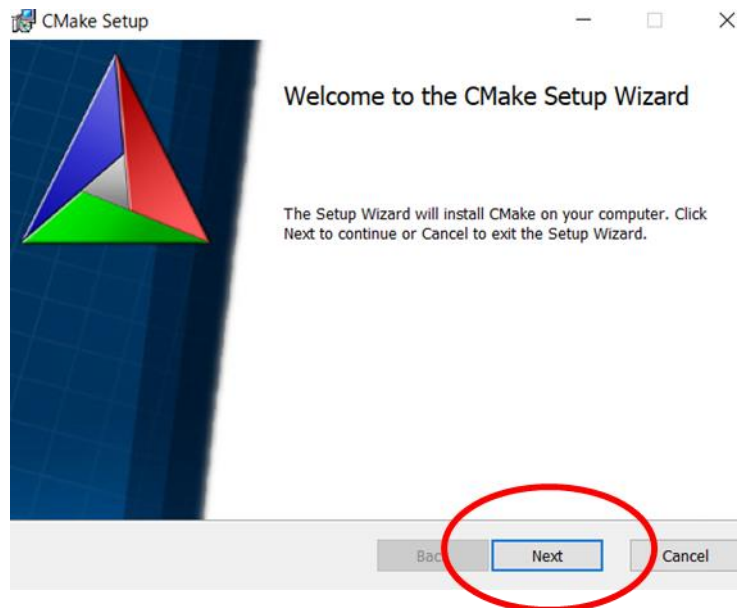


図 42 CMake 起動画面

3. 図 43 のチェックボックスにチェックを入れ, 「Next」をクリックする.

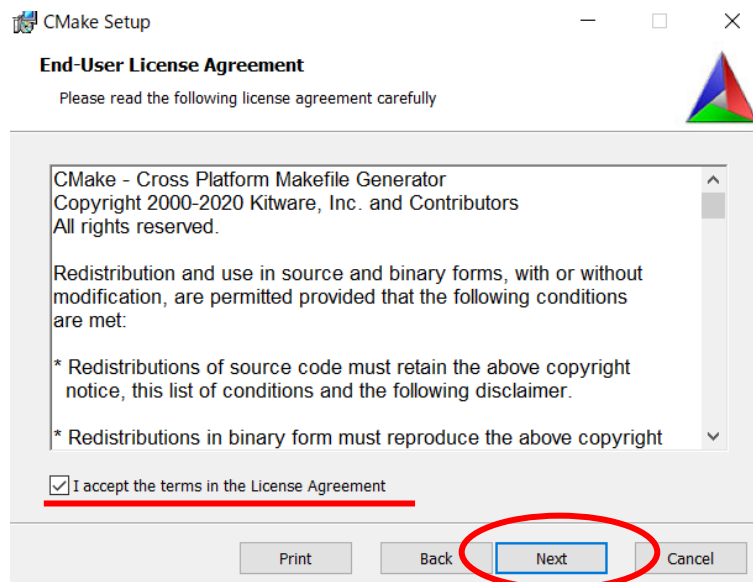


図 43 CMake 同意書画面

4. 環境変数に PATH を追加するために, 真ん中の「Add CMake to the system PATH for all users」にチェックを入れ, 「Next」をクリックする. なお, デスクトップ上に CMake の

ショートカットを作成したい場合、「Create CMake Desktop Icon」にチェックを入れる。

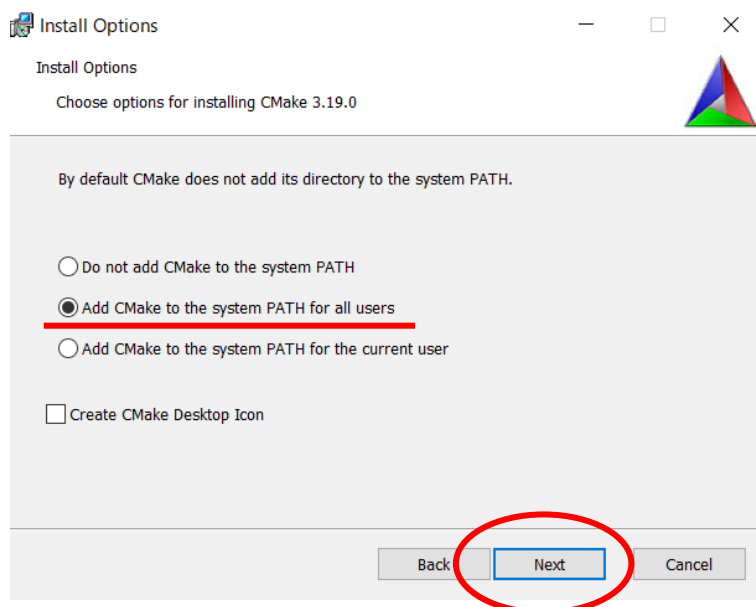


図 44 パスの設定

5. インストールしたいフォルダを指定し、「Next」をクリックする。基本デフォルト（「C:\Program Files\CMake\」）のままでよい。

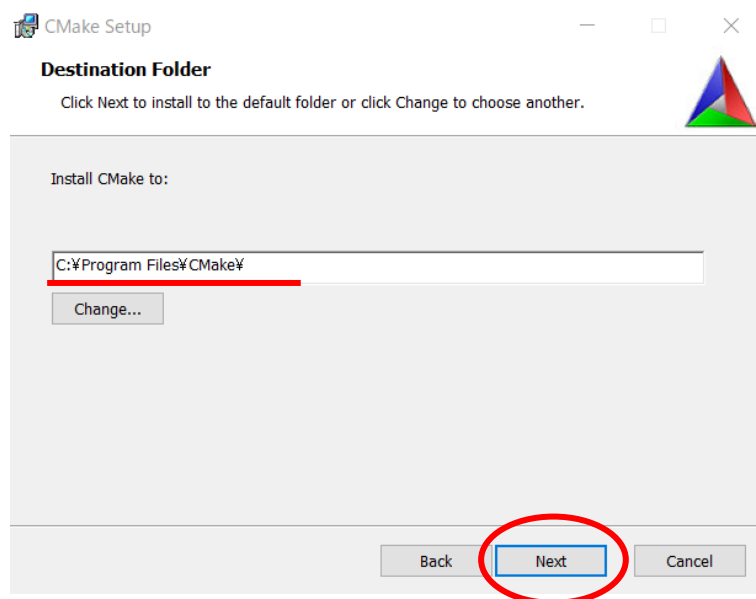


図 45 パスの設定

6. この画面になったら, 「Install」をクリックする. インストール完了後, 「Finish」をクリックする.

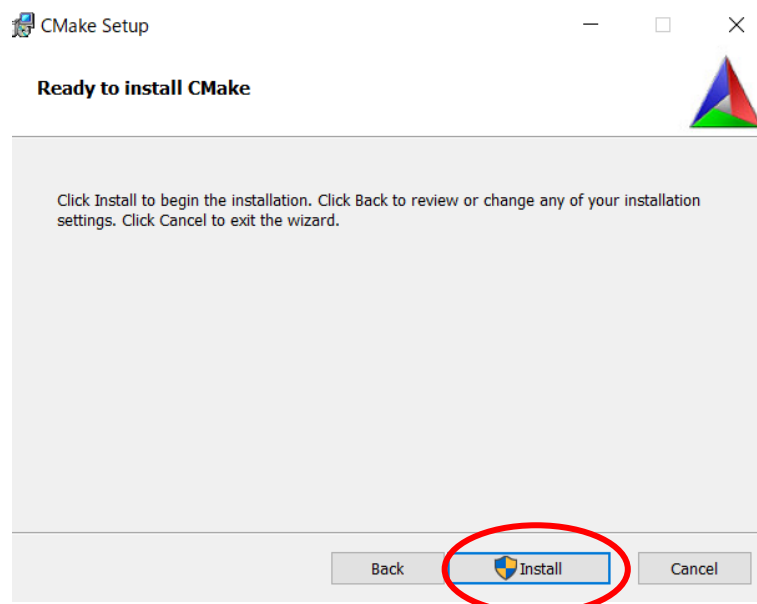


図 46 インストール完了画面

参考文献

- [1] International Council of Nurses: “ICN calls for data on healthcare worker infection rates and deaths,” <<https://www.icn.ch/news/icn-calls-data-healthcare-worker-infection-rates-and-deaths>>, (参照日: 2020 年 11 月 10 日)
- [2] Yang et al: “Combating COVID-19-The role of robotics in managing public health and infectious disease”, SCIENCE ROBOTICS EDITORIAL 5, pp.1-2(2020)
- [3] DX ライブラリ置き場, <https://dxlib.xsrv.jp/function/dxfunc_input.html>, (参照日: 2020 年 11 月 10 日)
- [4] DataTypeManual, <<https://nobu19800.github.io/DataTypeManual/docs/ExtendDataTypes#timedvelocity2d>>, (参照日: 2020 年 11 月 10 日)

連絡先

芝浦工業大学大学院 理工学研究科 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 : md20005<at>shibaura-it.ac.jp (飯田悠貴)

matsuhir<at>shibaura-it.ac.jp (松日楽信人)