

ユーザーズマニュアル 鳥瞰視点カメラロボット (ドローン) ミニサーベイヤーシステム

発行日 2017年3月22日
公立大学法人会津大学
株式会社東日本計算センター

目次

1. はじめに.....	1
1.1. 鳥瞰視点ロボットとは	1
1.2. 用語集.....	1
1.3. 略語集.....	2
1.4. 動作環境	2
1.4.1. 地上局 (PC)	2
1.5. 使用機器	3
1.6. 関連資料.....	3
2. 本システムでできること	4
3. 本システムのユーザーインターフェースについて	5
3.1. ドローンコントローラー仕様.....	5
3.1.1. PS4 コントローラー仕様詳細.....	5
3.2. ビューア	6
3.2.1. ドローンビューア	6
3.2.2. マップビューア	9
4. システム配置図.....	10
5. システム構成.....	11
6. システムの導入.....	12
6.1. インストール.....	12
6.1.1. ドローン.....	12
6.1.1.1. Z-Emulator	12
6.1.2. 地上局	13
6.1.2.1. OpenRTM-aist, OpenRTP	13
6.1.2.2. PyYAML, CMake, Doxygen.....	13
6.1.2.3. CoreAPI.....	13
6.1.2.4. kivy	17
6.1.2.5. mapview	17
6.2. 起動.....	18
6.2.1. ドローン.....	18
6.2.2. 地上局	18
6.2.2.1. 事前準備.....	18
6.2.2.2. Game Controller RTC.....	18
6.2.2.3. ドローンコントローラーRTC.....	20
6.2.2.4. MiniSV06LA RTC.....	20

6.2.2.5. ドローンマップビューア RTC.....	20
6.2.3. 全 RTC 接続.....	21
7. 各種設定.....	26
7.1. MiniSV06LA RTC.....	26
7.2. ドローンコントローラーRTC.....	26
7.3. ドローンマップビューア RTC.....	27
8. エラーメッセージ.....	28
9. FAQ.....	30

1. はじめに

1.1. 鳥瞰視点ロボットとは

鳥瞰視点ロボット（以下ドローン）は、無人航空機の事を指し、本書は(株)自律制御システム研究所製の6発ローターのマルチコプター「ミニサーベイヤー(MS06LA)」を用いたシステムのユーザーマニュアルです。本システムは平成28年度 会津大学 ロボットバレー創出推進事業の一環として開発したものです。なお、28年度はシミュレーター上でのフライト制御となります。

1.2. 用語集

本書で用いる用語を一覧に示します。

表 1-1 用語一覧

用語	読み	説明
Armed	アームド	ドローン飛行可能状態
disarm	ディスアーム	ドローン飛行不可状態
X-Monitor	エックスモニター	ミニサーベイヤーの基地局ソフトウェア
Y-Planner	ワイプランナー	ミニサーベイヤーの飛行計画ソフトウェア
Z-Emulator	ゼットエミュレーター	ミニサーベイヤーのエミュレーター
マニュアル	マニュアル	フライトモードの一種。GPS 未受信での機体操作。
姿勢アシスト	シセイアシスト	フライトモードの一種。GPS 未受信で姿勢角を維持。
GPS アシスト	ジーピーエスアシスト	フライトモードの一種。GPS 受信により安定した飛行(プロポのみ)が可能。
自律ナビゲーション	ジリツナビゲーション	フライトモードの一種。X-Monitor を使用して自律飛行を行う。
コマンド飛行	コマンドヒコウ	自律ナビゲーションの詳細モード。ターゲット位置へ飛行する。
ゴーホーム	ゴーホーム	自律ナビゲーションの詳細モード。ホーム位置(高度は 10m)へ飛行する。GPS アシストモードまたは自律ナビゲーションモード時のみ動作可能。

自動離陸	ジドウリリク	フライトモードの一種。GPS アシストモードまたは自律ナビゲーションモード時のみ動作可能。
自動着陸	ジドウチャクリク	フライトモードの一種。現在の位置へ着陸。GPS アシストモードまたは自律ナビゲーションモード時のみ動作可能。

1.3. 略語集

本書で用いる略語を一覧に示します。

表 1-2 略語一覧

略語	説明
FC	フライトコントローラー
PS4	PlayStation4
自律研	(株)自律制御システム研究所
ACSL	
RTL	Return to launch (ゴーホーム+自動着陸)

1.4. 動作環境

動作環境一覧は以下のとおりです。

1.4.1. 地上局 (PC)

表 1-3 地上局環境一覧

	環境	バージョン	補足
OS	Windows 10 Pro	1703	1511,1607 も動作可
CPU	Intel® Core™ i7-6650U CPU @ 2.20GHz 2.21GHz	-	-
メモリ	16GB	-	-
ストレージ	SSD 256GB	-	-
RT ミドルウェア	OpenRTM-aist	1.1.2	C++, Python, Java
依存ライブラリー	CoreAPI	0.0.8	-
	Kivy	1.9.1	-

1.5. 使用機器

表 1-4 使用機器一覧

No	機器名称	個数	補足
1	Surface Pro 4	1	本システムの基地局 PC ミニサーベイヤ-SDK のインストール要
2	hp Z840 Workstation	1	本システムのドローンエミュレーター PC Z-Emulator のインストール要
3	プロポ	1	ドローンの操縦装置(Futaba 製 10J)
4	プロポ受信機	1	自律研製 Z-Link と Futaba 製 R3008SB との 組み合わせ
5	USB<->ミニ USB ケーブル	1	Z-Link と Z-Emulator 用 PC を接続するケーブル
6	920MHz 無線モジュール	2	Futaba 製 FEP-01。基地局<->ドローンとの 通信に使用。各 PC の USB ポートに挿入。ド ライバインストール要。
7	DUAL SHOCK 4	1	旧型(CUH-ZCT1J)の PS4 コントローラー
8	USB ハブ	1	基地局側の USB ポートが 1 口のため、本機器 にてポートを拡張

1.6. 関連資料

表 1-5 資料一覧

No	資料名	補足
1	機能仕様書_MiniSV06LA	https://rtc-fukushima.jp/component/1775/
2	機能仕様書_DroneViewer	https://rtc-fukushima.jp/component/1031/
3	機能仕様書_DroneController	https://rtc-fukushima.jp/component/1175/
4	RTC 概要_RTC_GameController	https://rtc-fukushima.jp/component/1139/
5	機能仕様書_DroneMapView	https://rtc-fukushima.jp/component/1776/
6	CoreAPI 取扱説明書	ミニサーベイヤ通信ライブラリー。Ver1.2。
7	X-Monitor 取扱説明書	Ver1.2.2
8	Y-Planner 取扱説明書	Ver1.2.1
9	Z-Emulator 取扱説明書	Ver2.0.0
10	ミニサーベイヤ取扱説明書	制御対象ドローン(MS-06LA)のマニュアル

2. 本システムでできること

①. 離陸

機体側で設定されている高度まで離陸します。離陸中の他操作は出来ません。

②. 着陸

着陸はドローン起動位置への帰還のみです。帰還中の他操作は出来ません。

③. 移動

現座標に対して（移動速度 × ターゲット経緯度算出係数）分の経緯度差分¹を加味したターゲット座標を設定し移動します。移動方向は東西南北となります。

ターゲット速度は 1m/s 固定です。

④. 高度上昇/下降

ホームポジションの高度から 2~15m(1m 刻み)まで変更可能です。高度下降による着陸は出来ません。

⑤. 表示

専用ビューアにセンサー情報等を表示します。詳細は 3.2 で後述します。

¹ 10m あたりの緯度差分: 0.000137 度、10m あたりの経度差分: 0.000108 度（会津での使用を想定）

3. 本システムのユーザーインターフェースについて

3.1. ドローンコントローラー仕様

本システムでは、PS4 コントローラーによる操作が基本ですが、従来通り(プロポ)の操作も可能です。

3.1.1. PS4 コントローラー仕様詳細

コントローラーのキー仕様は下表のとおりです。

キーが押下されている限り動作は継続します。相反する操作(例：高度上昇/下降を同時に要求)をした場合は何もしません。**MS-06LA の離陸はプロポで行う仕様ですのでキーによる離陸は出来ません。**

なお、ドローンコントローラーRTC の動作周期は 1 秒であるため、**動作周期以下のキー状態変化検知は行いません** (例：離陸指示する場合には R1 キーを 1 秒間押し続けます)。

表 3-1 キーマッピング一覧

キー名称	動作
十字キー上	移動(北)
十字キー下	移動(南)
□	移動(西)
○	移動(東)
△	高度上昇
×	高度下降
R2	帰還
OPTIONS	移動モード切り替え
上記以外	—

3.2. ビューア

3.2.1. ドローンビューア

以下にドローンビューアおよび各表示項目について説明します。
灰色網掛け表示部分(表 3-2)は、本システムでは表示しません。

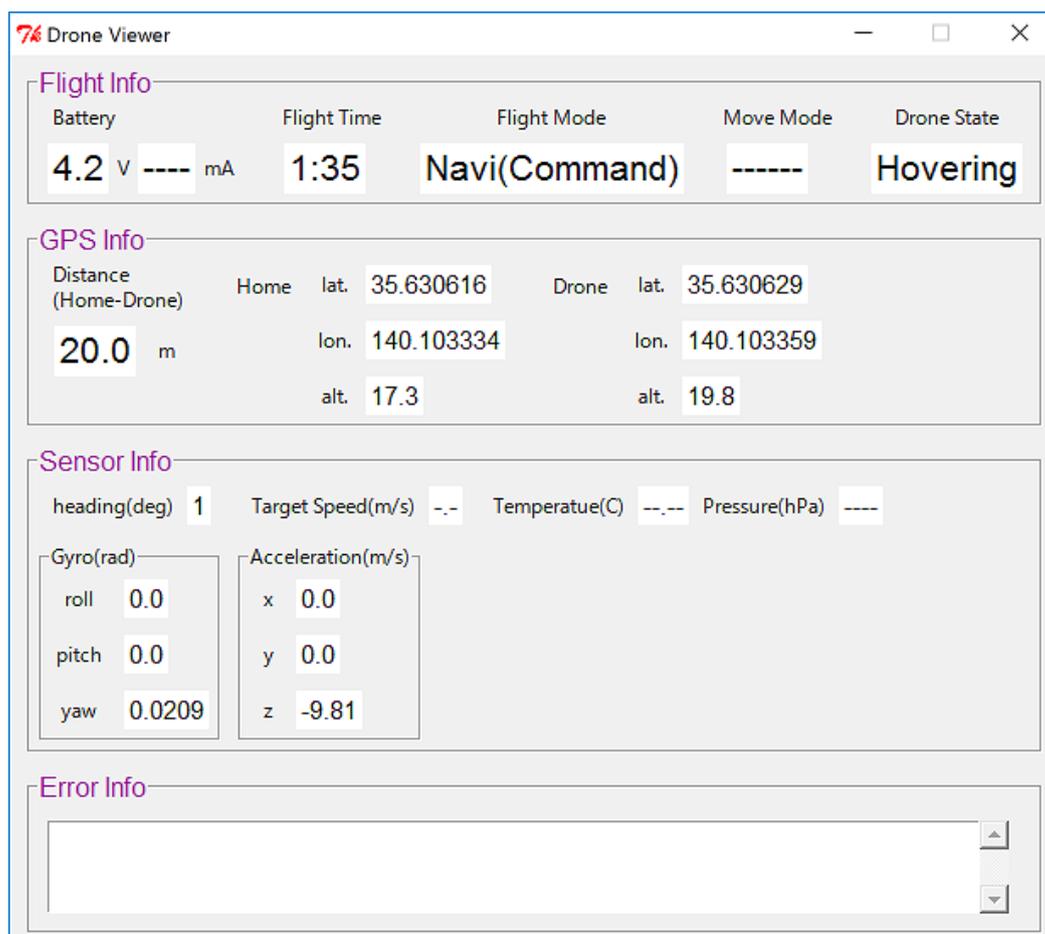


図 3-1 ドローンビューア

表 3-2 ドローンビューア表示項目一覧

項目	単位	説明																		
Battery	V	バッテリーの電圧値																		
	mA	バッテリーの電流値																		
Flight Time	分秒	フライト時間																		
Flight Mode	-	<p>現在のフライトモードを表示</p> <p>本システムのフライトモードはミニサーベイヤーのコントロールモード及び自律モードを統合。</p> <table border="1"> <thead> <tr> <th>表示仕様</th> <th>意味</th> </tr> </thead> <tbody> <tr> <td>Manual</td> <td>マニュアル</td> </tr> <tr> <td>Assist(Gyro)</td> <td>姿勢アシスト</td> </tr> <tr> <td>Assist(GPS)</td> <td>GPS アシスト</td> </tr> <tr> <td>Navi(Command)</td> <td>自律(コマンド飛行)</td> </tr> <tr> <td>Navi(GoHome)</td> <td>自律(ゴーホーム)</td> </tr> <tr> <td>Auto-takeoff</td> <td>自動離陸</td> </tr> <tr> <td>Auto-landing</td> <td>自動着陸</td> </tr> </tbody> </table>	表示仕様	意味	Manual	マニュアル	Assist(Gyro)	姿勢アシスト	Assist(GPS)	GPS アシスト	Navi(Command)	自律(コマンド飛行)	Navi(GoHome)	自律(ゴーホーム)	Auto-takeoff	自動離陸	Auto-landing	自動着陸		
表示仕様	意味																			
Manual	マニュアル																			
Assist(Gyro)	姿勢アシスト																			
Assist(GPS)	GPS アシスト																			
Navi(Command)	自律(コマンド飛行)																			
Navi(GoHome)	自律(ゴーホーム)																			
Auto-takeoff	自動離陸																			
Auto-landing	自動着陸																			
Move Mode	-	<p>現在の移動モードを表示</p> <table border="1"> <thead> <tr> <th>モード</th> <th>意味</th> </tr> </thead> <tbody> <tr> <td>Absolute</td> <td>東西南北移動</td> </tr> <tr> <td>Relative</td> <td>機体の向きに対する移動</td> </tr> </tbody> </table>	モード	意味	Absolute	東西南北移動	Relative	機体の向きに対する移動												
モード	意味																			
Absolute	東西南北移動																			
Relative	機体の向きに対する移動																			
Drone State	-	<p>ミニサーベイヤーRTC の内部モードを表示</p> <table border="1"> <thead> <tr> <th>モード</th> <th>意味</th> </tr> </thead> <tbody> <tr> <td>Disarm</td> <td>ドローン飛行不可能状態</td> </tr> <tr> <td>Armed</td> <td>ドローン飛行可能状態</td> </tr> <tr> <td>Takeoff</td> <td>離陸中</td> </tr> <tr> <td>RTL</td> <td>帰還中</td> </tr> <tr> <td>In_Move</td> <td>移動中</td> </tr> <tr> <td>Alt_CHG</td> <td>高度変更中</td> </tr> <tr> <td>Hovering</td> <td>ホバリング中</td> </tr> <tr> <td>Yaw_CHG</td> <td>旋回中</td> </tr> </tbody> </table>	モード	意味	Disarm	ドローン飛行不可能状態	Armed	ドローン飛行可能状態	Takeoff	離陸中	RTL	帰還中	In_Move	移動中	Alt_CHG	高度変更中	Hovering	ホバリング中	Yaw_CHG	旋回中
モード	意味																			
Disarm	ドローン飛行不可能状態																			
Armed	ドローン飛行可能状態																			
Takeoff	離陸中																			
RTL	帰還中																			
In_Move	移動中																			
Alt_CHG	高度変更中																			
Hovering	ホバリング中																			
Yaw_CHG	旋回中																			
Distance	m	ホームポジション(ドローン起動位置)とカレントポジション(ドローン)の距離																		

Home Drone	lat.	rad.	ホームポジションとカレントポジションの緯度/経度/高度。ホームポジションの高度は海拔高度、カレントポジションの高度はホームポジションの高度からの相対値。
	lon.		
	alt.	m	
heading		deg.	磁気コンパス 北を 0° とし時計回りに角度を表示。
Gyro		rad.	3 軸ジャイロセンサー
Target Speed		m/sec	ターゲット速度
Acceleration		m/sec	3 軸加速度センサー
Temperature		C	気温
Pressure		hPa	気圧
Error Info		-	ミニサーベイヤ-RTC のエラー情報を表示

3.2.2. マップビューア

ドローンの位置情報を地図上(Open Street Map)に表示します。

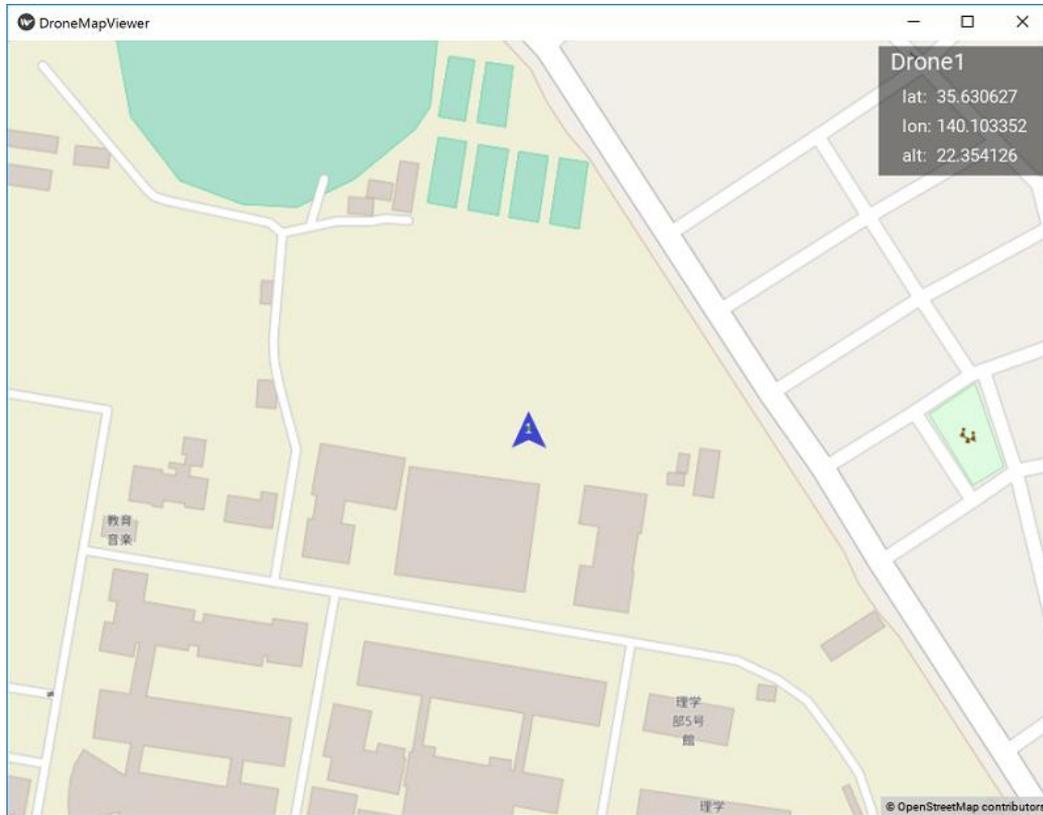


図 3-2 ドローンマップビューア

4. システム配置図

上がドローン(=シミュレーター)、下が地上局です。

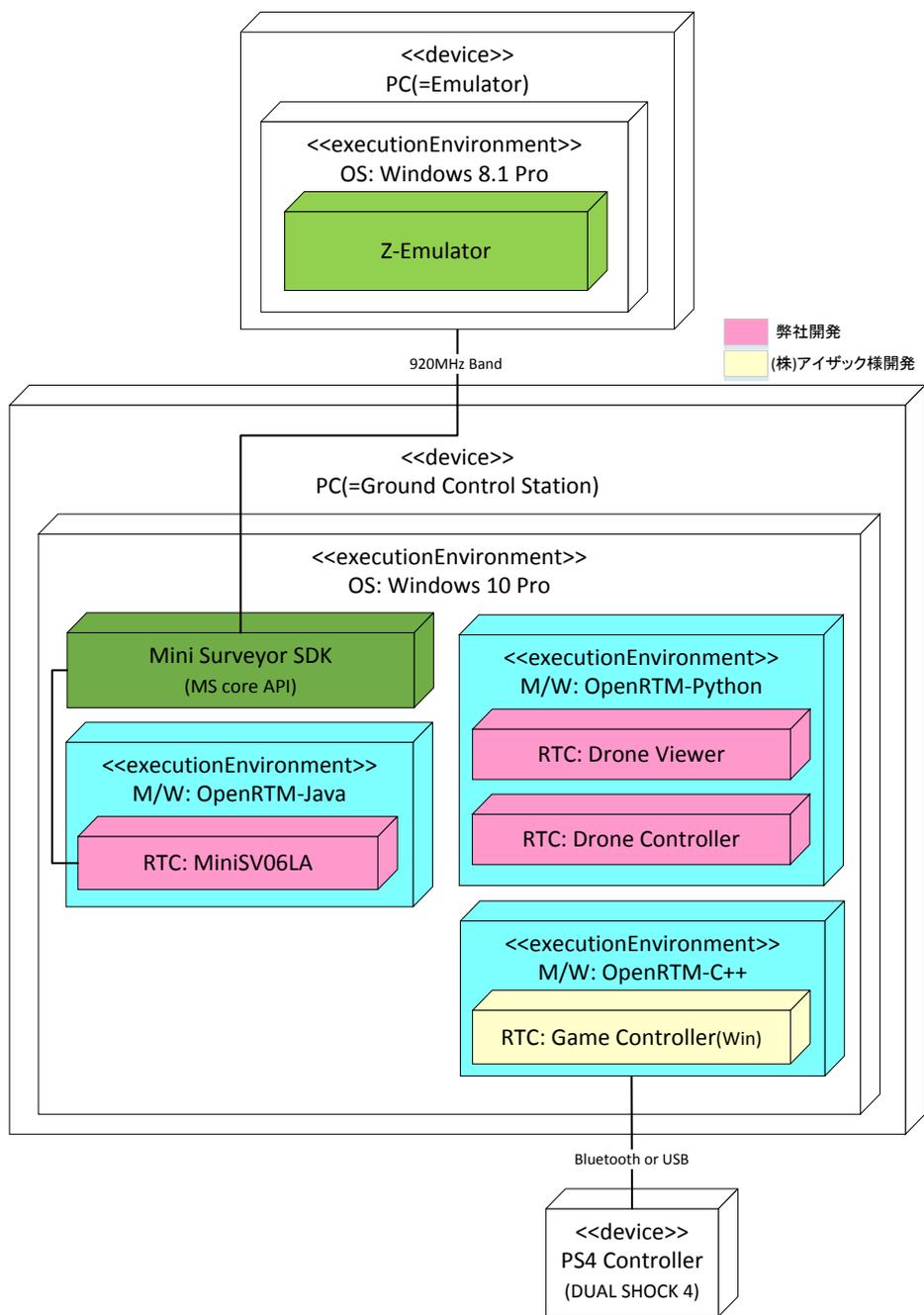


図 4 - 1 システム配置図

5. システム構成

表 5-1 システム構成一覧

ディレクトリ名	ファイル名	説明
MiniSV06LA	MiniSV06LA.java	実行ファイル
	MiniSV06LAComp.java	
	MiniSV06LAImpl.java	
	DroneMathmetics.java	2点間距離算出ファイル
	MiniSV06LA.conf	コンフィギュレーションファイル
	rtc.conf	
	RTC.xml	プロファイル
DroneController	DroneController.py	実行ファイル
	target_gps.py	ターゲット GPS 算出
	drone_distance_cal.py	2点間距離算出ファイル
	DroneController.conf	コンフィギュレーションファイル
	rtc.conf	
	RTC.xml	プロファイル
DroneViewer	DroneViewer.py	実行ファイル
	create_sensor_panel.py	ビューア本体ファイル
	drone_distance_cal.py	2点間距離算出ファイル
	DroneViewer.conf	コンフィギュレーションファイル
	rtc.conf	
	RTC.xml	プロファイル
RTC_GameController	/include/RTC_GameController.h	ヘッダファイル
	/src/RTC_GameController.cpp	ソースファイル
	RTC_GameController.conf	コンフィギュレーションファイル
	rtc.conf	
	RTC.xml	プロファイル
DroneMapView	DroneMapView.py	実行ファイル
	map_create.py	画面生成ファイル
	DroneMapView.conf	コンフィギュレーションファイル
	rtc.conf	
	RTC.xml	プロファイル

6. システムの導入

6.1. インストール

RTC-Library-FUKUSHIMA から以下の RTC をダウンロードし、任意のディレクトリに配置し解凍します。

表 6-1 ドローンシステム パス一覧

名称	パス
MiniSV06LA	https://rtc-fukushima.jp/component/1775/
ドローンコントローラー	https://rtc-fukushima.jp/component/1175/
ドローンビューア	https://rtc-fukushima.jp/component/1031/
RTC_GameController	https://rtc-fukushima.jp/component/1139/
ドローンマップビューア	https://rtc-fukushima.jp/component/1776/

6.1.1. ドローン

6.1.1.1. Z-Emulator

- (1) Z-Emulator 用 PC にプロポ受信機(Z-Link)ならびに 920MHz 無線モジュールを接続してください。
- (2) 後述しますが、Z-Emulator にて COM ポート設定が必要です。Z-Link と無線モジュールの COM ポート番号をデバイスマネージャー画面で確認します。



図 6-1 デバイスマネージャー(ポート)

- (3) Z-Emulator 取扱説明書にしたがい、Z-Emulator をインストールしてください。
- (4) インストール完了後、Z-Emulator 取扱説明書にしたがい、動作することを確認してください。

6.1.2. 地上局

6.1.2.1. OpenRTM-aist, OpenRTP

以下のページからインストールしてください。

<http://openrtm.org/openrtm/ja/node/6034>

6.1.2.2. PyYAML, CMake, Doxygen

Game Controller RTC のビルドに必要なになります。

以下のページからインストールしてください。

<http://openrtm.org/openrtm/ja/node/6034>

6.1.2.3. CoreAPI

(1) RTC Builder にて、MiniSV06LA RTC をインポートしてください。

(2) CoreAPI(minisv-distribution-0.0.8-SNAPSHOT-release.zip)を任意の場所で解凍してください。以降、minisv-distribution-0.0.8-SNAPSHOT-release をルートフォルダ(¥¥)として説明します。

(3) 参照ライブラリーのコンテキストメニューより、「ビルド・パス」-->「ビルド・パスの構成」を選択してください。

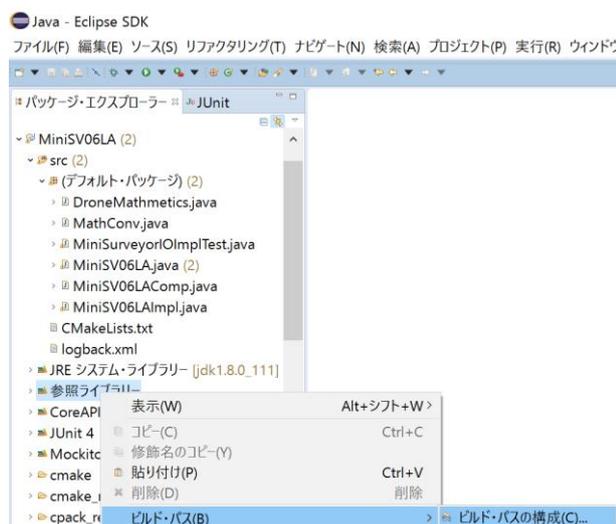


図 6-2 参照ライブラリー

(4) [ライブラリー]タブから「外部 JAR の追加」を選択してください。

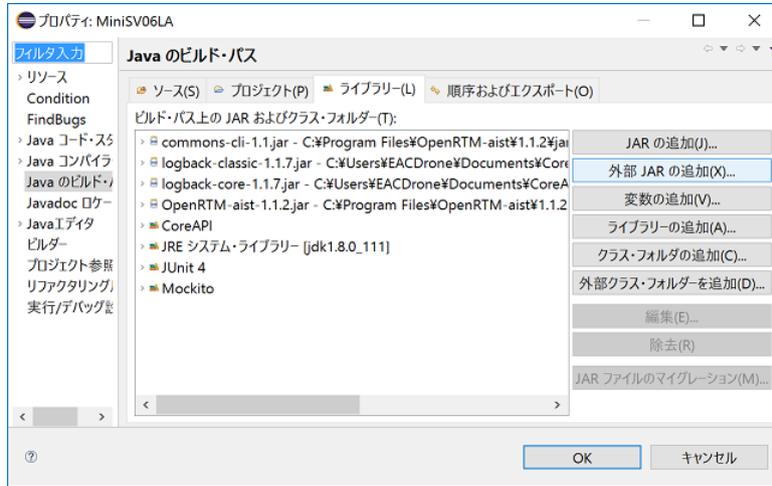


図 6-3 ビルド・パス設定(ライブラリー)

(5) %lib 内のライブラリーを全て追加してください。

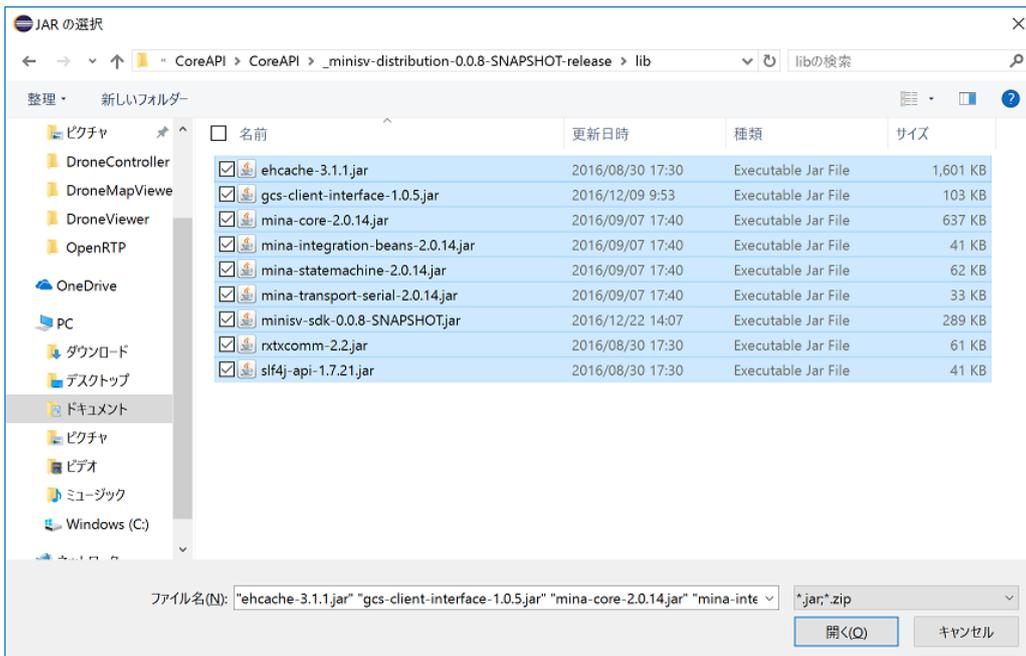


図 6-4 Core API 各種ライブラリー

- (6) 再度(3)を実施します。
- (7) C:\Users\ユーザー名\.m2\repository\ch\qos\logback 内の jar を全て追加してください。追加しなくても実行可能ですが、SLF4J のエラーが出力されます。

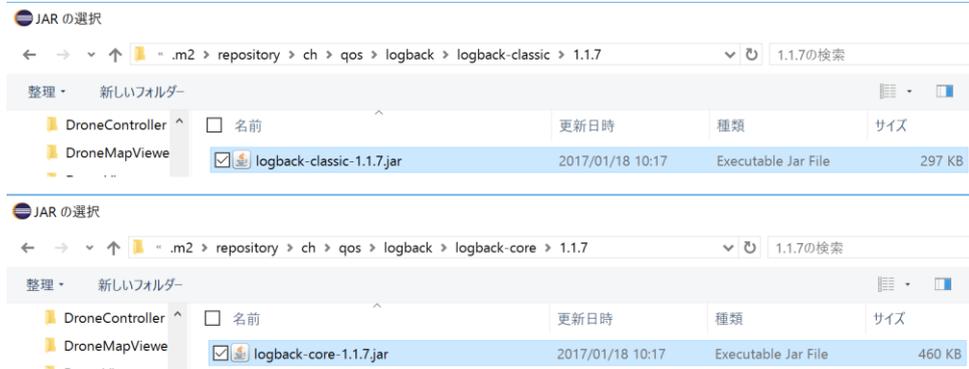


図 6-5 logback 関連 JAR ファイル

- (8) %sample%\minisv-sample-client\src\main\resources\内の logback.xml をコピーする。
- (9) (8)をコード生成した src と bin フォルダ内にペーストする。これにより、コンソール上に logback に関する DEBUG レベルの log は出力されません。

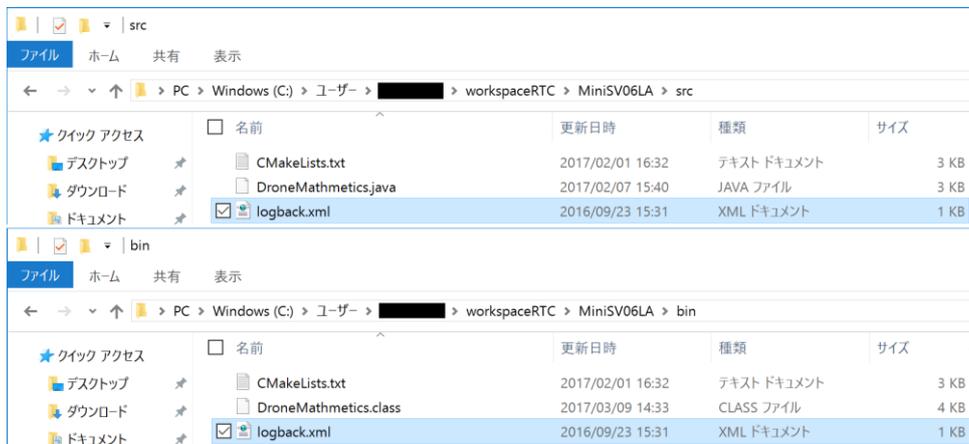


図 6-6 logback コピー

- (10) 再度(3)を実施してください。

- (11) [ライブラリー]タブ内の「JRE システム・ライブラリー」を展開し、「ネイティブ・ライブラリー・ロケーション」を選択し、「編集」を選択します。

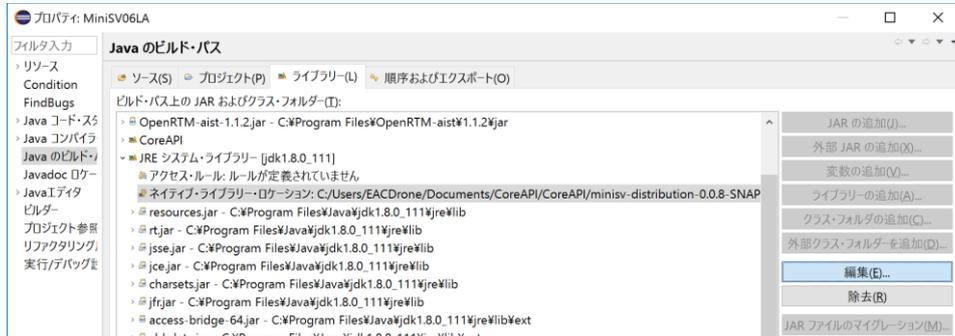


図 6-7 ネイティブ・ライブラリー・ロケーション編集(1)

- (12) %ext%rxtx-2.2%win-x64 フォルダを選択します。

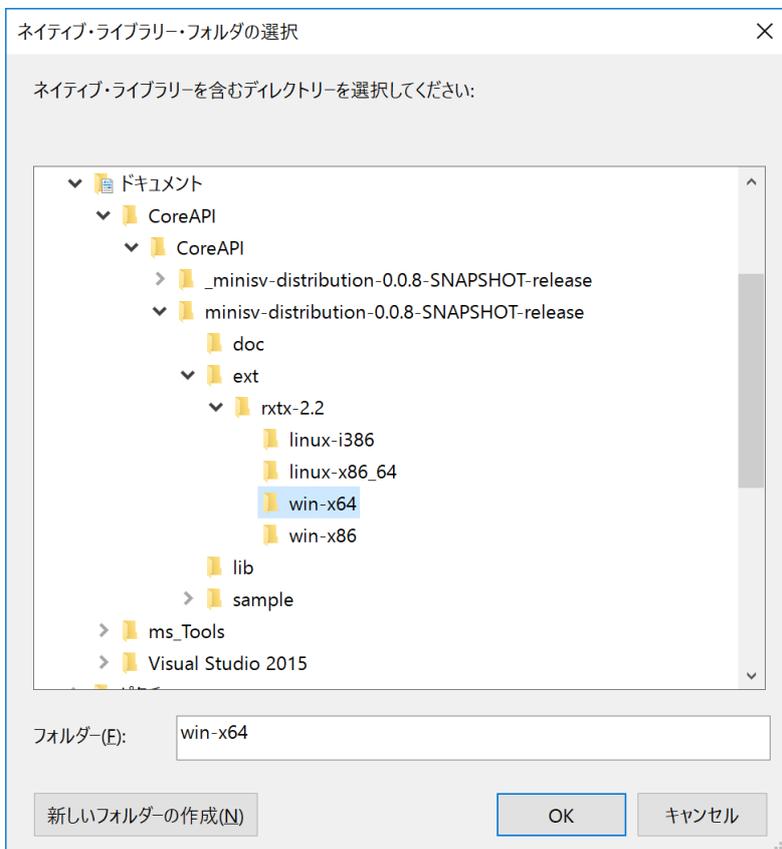


図 6-8 ネイティブ・ライブラリー・ロケーション編集(2)

- (13) パッケージ・エクスプローラー内の「build_(コンポーネント名).xml」のコンテキストメニューより、「実行」-->「Ant ビルド」を選択します。ビルド失敗した場合は JRE のバージョンが 1.8.0 かどうかを確認ください。

6.1.2.4. kivy

ドローンマップビューア RTC は kivy を用いて実装していますので、本パッケージ(安定版)をコマンドプロンプト(管理者)からインストールします。kivy_examples はオプション(デモやサンプルコード等)ですので必要に応じてインストールください。

```
python -m pip install --upgrade pip wheel setuptools
python -m pip install docutils pygments pypiwin32 kivy.deps.sdl2 kivy.deps.glew
python -m pip install kivy
python -m pip install kivy_examples
```

6.1.2.5. mapview

kivy 上での地図表示要件を実現するために必要になります。mapview は kivy-garden(kivy のアドオン)として提供されていますので、kivy-garden のインストールも必要になります。さらに mapview 必要要件モジュールとして concurrent.futures と requests も必要になります。

```
pip install futures requests
pip install kivy-garden
garden install mapview
```

6.2. 起動

6.2.1. ドローン

Z-Emulator を「管理者として実行」します。飛行条件は以下の内容で設定します。

表 6-2 Z-Emulator 設定一覧

飛行場所	千葉大学
ドローンの種類	MS-06LA
霧	無し
太陽光	有り
風	無し
追跡	無し
タイマー	120 分
レーザーモジュールのポート設定	図 6-1 参照(COM5)
920MHz モジュールのポート設定	図 6-1 参照(COM4)

6.2.2. 地上局

6.2.2.1. 事前準備

(1) ネームサーバー(Start Naming Service)を起動します。

6.2.2.2. Game Controller RTC

以下の手順(1)~(4)は PS4 コントローラーを**無線接続**する場合に設定が必要です。

(1) タスクバーの通知領域内の Bluetooth アイコンを右クリックし、「Bluetooth デバイスの表示」を選択します。

(2) PS4 コントローラーの電源を ON (PS ボタン押下) します。ペアリング未設定の場合は、ペアリングモード (SHARE ボタンと PS ボタン長押し) に切り替えます。切り替わると、コントローラー上部のライトバーが白色点滅します。

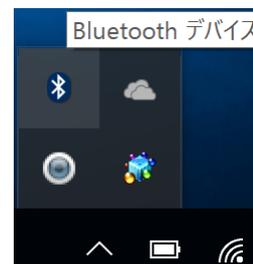


図 6-9 通知領域

(3) 白色点滅中に、Wireless Controller --> ペアリングの順で選択し、ライトバーが青色点灯に切り替わったら、ペアリング成功です。次回以降コントローラーを使用する場合は、PS ボタン押下のみでペアリングします。



図 6-1 0 Bluetooth 設定画面

(4) PS4 コントローラーのプロパティにて、各種ボタンが効くかどうかをご確認ください。

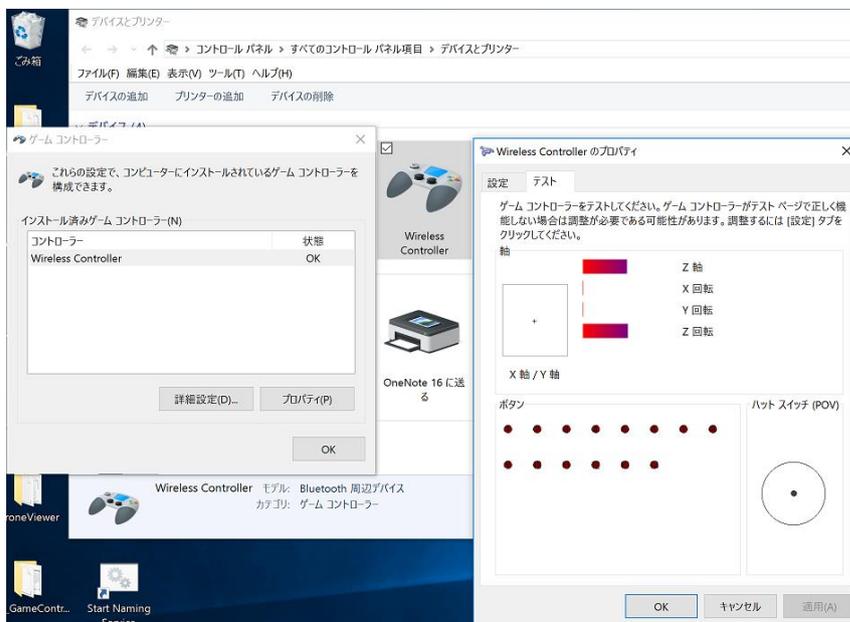


図 6-1 1 ゲームコントローラー プロパティ画面

(5) ビルド方法、実行方法は 1.4. 関連資料の No.4 を参照ください。

6.2.2.3. ドローンコントローラーRTC

(1) ドローンコントローラーRTCの実行ファイルを起動します。

6.2.2.4. MiniSV06LA RTC

(1) パースペクティブをJavaに切り替えます。

(2) メニューバーの「実行」-->「実行」(またはツールバー内の実行ボタン)を選択することで、MiniSV06LA RTCを起動します。コンソールに「テレメトリーの取得設定に成功」が表示されましたら、起動成功です。

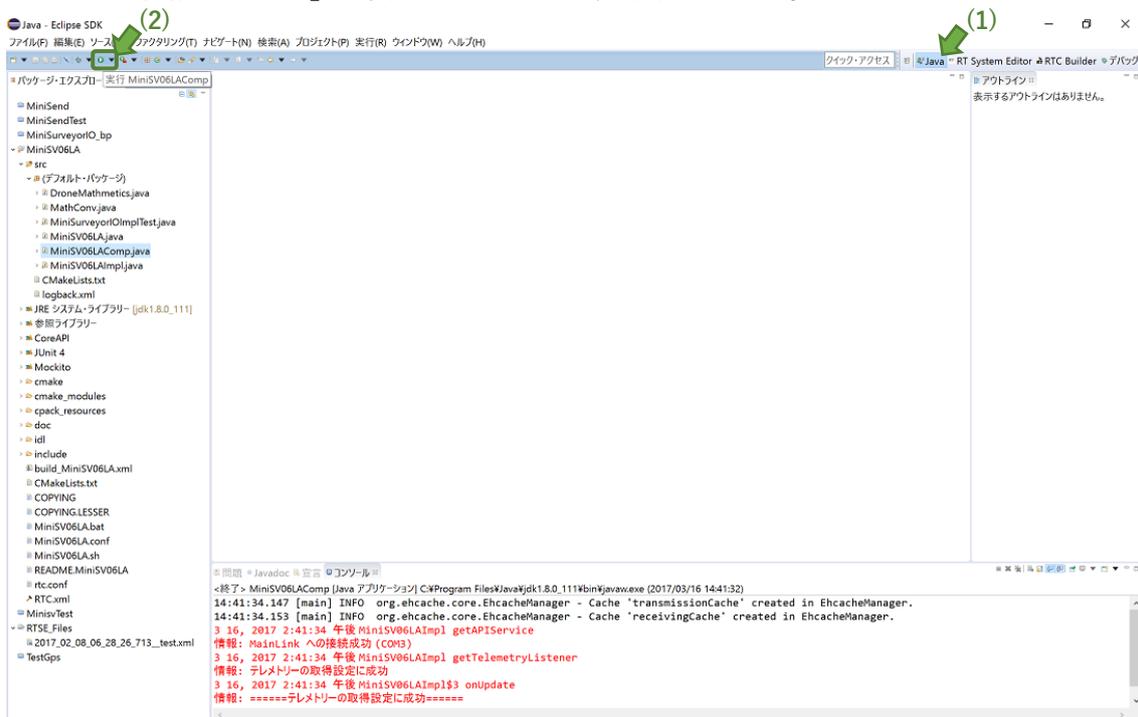


図 6-1 2 MiniSV06LA RTC 起動手順

6.2.2.5. ドローンマップビューア RTC

(1) ドローンマップビューア RTCの実行ファイルを起動します。

6.2.3. 全 RTC 接続

- (1) OpenRTP を起動し、「RT System Editor」 パースペクティブに切り替えます。
- (2) ネームサーバーの追加（地上局）を行います。
- (3) 全 RTC を System Diagram にセットします。
- (4) Drone Controller の動作周期を変更(250msec-->1sec)します。Drone Controller を選択し、Execution Content View にて、4Hz-->1Hz に変更します。

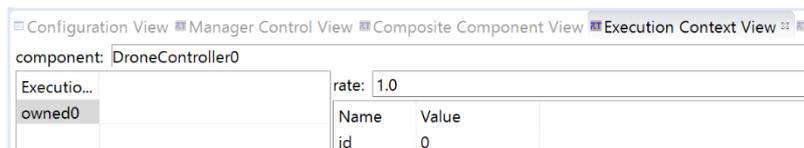


図 6-1 3 Drone Controller の Execution Content View

- (5) 各 RTC のコンフィグパラメータの設定を行います。詳細は 7.各種設定を参照ください。ドローンコントローラーは飛行場所の海拔高度に応じて MinAlt と MaxAlt の値を変更する必要があります。
- (6) RTC 間のポート接続方法は以下のとおりです。

a. RTC_GameController

各ポートとも、Connector Profile の変更は以下のみ。Controller_Type は本システムでは使用しません。

Dataflow Type : pull Buffer(Output)の Buffer length : 1

(a) RTC_GameController.Button -> DroneController.ButtonIn

(b) RTC_GameController.Analog -> DroneController.AnalogIn

b. DroneController

各ポートとも、Connector Profile の変更は以下のみ。

Buffer(Inport)の Buffer length : 1

(a) DroneController.TargetGPSOut -> MiniSV06LA.inTargetPos

(b) DroneController.TargetGPSOut -> DroneMapView.Target1GPSIn

c. MiniSV06LA

各ポートとも、Connector Profile の変更は以下のみ。

Buffer(Inport)の Buffer length : 1

- (a) MiniSV06LA.outGPSHome -> DroneController.HomeGPSIn
- (b) MiniSV06LA.outGPSDrone -> DroneController.DroneGPSIn
- (c) MiniSV06LA.outGPSDrone -> DroneMapViewer.Drone1GPSIn
- (d) MiniSV06LA.outMag -> DroneMapViewer.inMag1
- (e) MiniSV06LA.outState -> DroneFlightDirector.inState1
- (f) MiniSV06LA -> DroneViewer

BattA, Temp, Pres, RawIMU、CorrectIMU, TargetSpeed, MoveMode は接続対象外。

接続完了しますと、以下ようになります。再度ご確認ください。

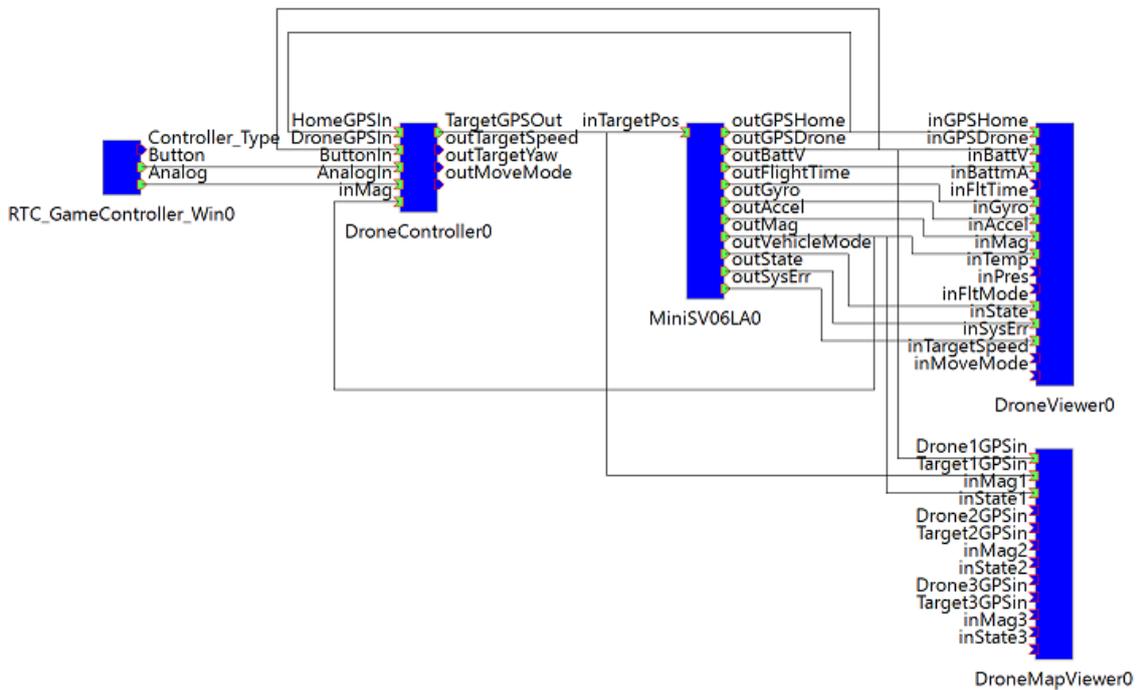


図 6-1 4 ドローンシステム接続完了時のシステムダイアグラム画面

接続完了後、システムエディタ（RTC 間の接続やコンフィギュレーション情報）を保存することをおすすめします。次回以降、保存したシステムエディタの内容を読み込むことで上記 a～d の手順を省くことができます。以下の手順で保存をします。

1. システムエディタウインドウ上でコンテキストメニューを開きます
2. Save（または Save as）を選択します
3. Profile Information が開く。Update log 以外は入力必須項目となる。入力し終えたら、OK を選択することで、保存が完了します

読み込みは以下の手順になります。

1. 読み込みに必要な RTC を起動し、ネームサーバー上に登録されていることを確認してください
2. システムエディタウインドウ上でコンテキストメニューを開きます
3. 保存内容を復元したい場合は Open and Restore、そうでない場合は Open を選択し、保存したファイル名を選択することで読み込みが完了します

- (7) プロポで離陸操作をするために、ドローンを Armed に切り替えます。プロポの電源を入れる前に、右側(エルロン/スロットル)スティックの位置がスロー(一番下)であるか、4 つの 2 ポジションスイッチが全て上に設定されているか、4 つの 3 ポジションスイッチが全て奥側に設定されているかを確認した後に、プロポの電源を入れます。



図 6-15 プロポ各種スイッチのポジショニング

- (8) 左(エレベーター/ラダー)スティックを数秒間左にすることで、Armed に切り替わります。Armed に切り替わったら(=モーターが動作したら)、スティックをニュートラルにします。切り替わらない場合は、Z-Emulator のリセットをした上で再度お試しください。



図 6-16 Armed 切り替え操作中(左)と Armed 切り替え完了(右)

- (9) 全 RTC を Activate(=All Activate)

以下のように、全て Active 状態に遷移することを確認ください。

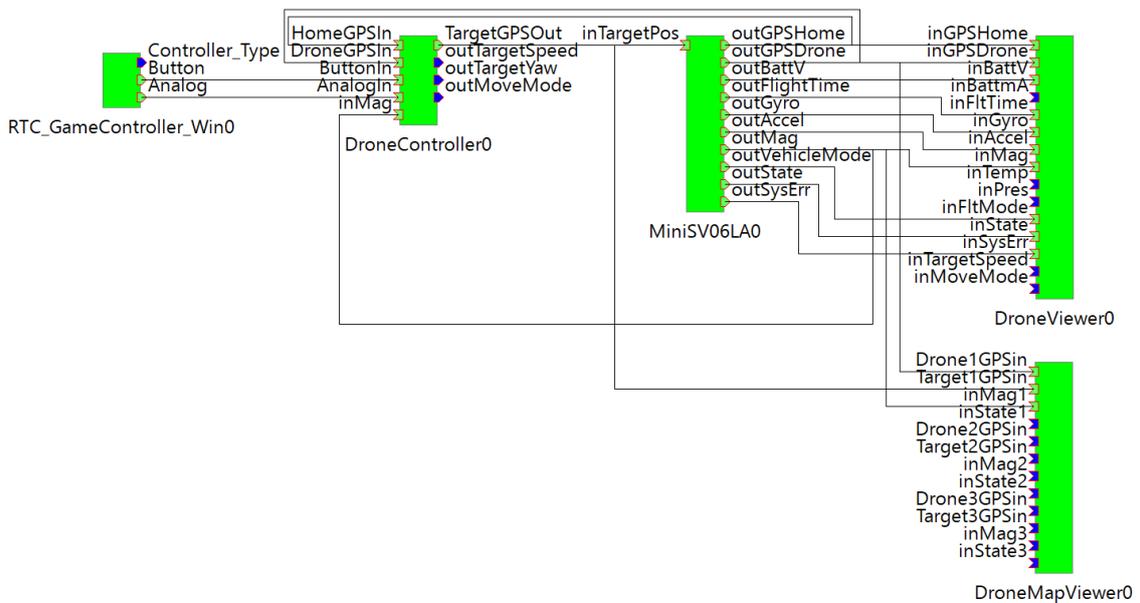


図 6-17 ドローンシステム 起動完了後のシステムダイアグラム画面

- (10) プロポの E スイッチを中央(GPS アシストモード)に切り替え、右スロットルの位置を中央にすることで離陸を開始します。



図 6-18 離陸操作

- (11) 離陸動作完了後、プロポの E スイッチを手前(自律ナビゲーションモード)に切り替えることで、PS4 コントローラーによるドローンの操作が可能です。操作方法は表 3-1 を参照ください。



図 6-19 ゲームコントローラーでの操作

7. 各種設定

各 RTC が持っているコンフィギュレーションによる調整機能について説明します。
 なお、灰色網掛け表示部は、このシステムでは使用しません。

7.1. MiniSV06LA RTC

表 7-1 MiniSV06LA RTC コンフィギュレーション一覧

名称	データ範囲	デフォルト値	説明
ConnectPort	-	COM3	920MHz 無線モジュールの接続 USBPort

7.2. ドローンコントローラ RTC

MinAlt は飛行場所のホームポジション海拔高度²に最小高度(2m)、MaxAlt は最大高度(15m)を加算した値を設定します。飛行場所に応じて変更してください。表 7-2 は千葉大学を飛行場所にした場合の設定値となります。それ以外は表 7-3 を参照ください。

表 7-2 ドローンコントローラ RTC コンフィギュレーション一覧

名称	データ範囲	デフォルト値	説明
AltCoefficient	$1 \leq x \leq 10$	1	ターゲット高度算出係数
LatLonCoefficient	$2 \leq x \leq 10$	10	ターゲット経緯度算出係数
LatDiff10m	$0.000000 < x < 0.100000$	0.0090133	1km あたりの緯度差分
LonDiff10m	$0.000000 < x < 0.100000$	0.0109664	1km あたりの経度差分
MoveSpeed	$1.0 \leq x \leq 10.0$	3.0	ドローン移動速度
MinAlt	$1.0 \leq x \leq 100.0$	19.2	ドローン最小高度
MaxAlt	$10.0 \leq x \leq 100.0^3$	32.2	ドローン最大高度
TurnSpeed	$0 \leq x \leq 360$	45	1秒あたりのドローン旋回速度
MoveTurnSpeed	$0 \leq x \leq 360$	30	1秒あたりの移動時の旋回速度
NoseLatDiff10m	$0.000000 < x < 10.000000$	※ ⁴	10m あたりの機首緯度差分
NoseLonDiff10m	$0.0000000 < x < 10.0000000$	※ ⁵	10m あたりの機首経度差分

² 事前に X-Monitor を起動することで確認可能。確認し終わったら、X-Monitor を終了する。

³ ドローン最大高度(MaxAlt)の設定値は、最小高度(MinAlt)の設定値以上の値とする。

⁴ デフォルト値： 9.01323×10^{-5} の「9.01323」の部分データをデータ範囲とする。

⁵ デフォルト値： $8.9831566 \times 10^{-5} / \cos(\text{lat}/2\pi)$ の「8.9831566」の部分データをデータ範囲とする。

表 7-3 飛行場所

飛行場所	海拔高度 [m]	MinAlt[m] (海拔高度+2m)	MaxAlt[m] (海拔高度+15m)
千葉大学	17.2	19.2	32.2
平地	17.2	19.2	32.2
建造物内部	18.4	20.4	33.4
秋葉原(東京)	4.2	6.2	19.2
廃墟	17.2	19.2	32.2
橋梁	17.2	19.2	32.2
トンネル	17.2	19.2	32.2
煙突	17.2	19.2	32.2
難波(大阪)	4.3	6.3	19.3
天神(福岡)	3.3	5.3	18.3
札幌(北海道)	18.9	20.9	33.9

7.3. ドローンマップビューア RTC

表 7-4 ドローンマップビューア RTC コンフィギュレーション一覧

名称	データ範囲	デフォルト値	説明
DroneNum	$1 \leq x \leq 3$	1	ドローン制御台数
MapZoomLevel	$0 \leq x \leq 19$	18	地図の縮尺レベル

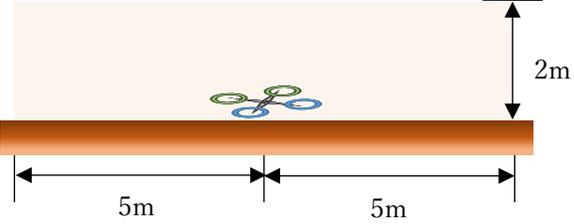
8. エラーメッセージ

エラー発生時、コンソール上にエラーメッセージを表示します。以下は各 RTC のエラーメッセージです。

表 8-1 MiniSV06LA RTC のエラーメッセージ一覧

No.	メッセージ	説明
1	[E]home_location is none. Please "Exit" this RTC.	ホームポジションのダウンロードに失敗しました。今回はシミュレーター上での動作ですので発生しません。
2	[E]home_location is not exist.Please "Reset" this RTC.	ホームポジションが存在しないまま、RTC を Activate した場合に発生します。
3	[E]No armed. Please "Reset" this RTC.	Armed 状態でないときに RTC を Activate した場合
4	[E]Communication with the GCS has been lost. Start the RTL.	ドローン<->地上局との通信が一定期間(3秒)途絶えた場合に発生します。ホームポジションに帰還します。帰還途中で通信良好となっても帰還動作し続けます。
5	指定された COM ポートに通信モジュールが接続されていません。通信モジュールを接続後、再起動してください。	指定 Port に 920MHz 無線モジュールが挿入されていない場合に発生します。

表 8-2 ドローンコントローラRTC のエラーメッセージ一覧

No.	メッセージ	説明
1	ERROR: xxx data is NOT available xxx: GPS または MAG	MiniSV06LA RTC から位置情報や磁気コンパスデータが受信できません。MiniSV06LA RTC が Active であることを確認してください。
2	WARNING: Conflicting operations are detected	相反する操作要求検知時に発生します。相反する操作要求を解除してください。
3	WARNING: Take-off operation is NOT available in the current drone position	<p>離陸前のポジションが誤差許容範囲内(ホーム〜ドローンポジション距離誤差が 5m 以内かつドローンポジションの高度誤差が 2m 以内)である場合は離陸可能ですが、これらの条件以外となった場合はすでに離陸完了とみなします。</p>  <p>GPS 誤差が大きいことで発生していますので、再度キャリブレーションをするか、FC や GPS アンテナそのものの交換が必要です。</p>
4	WARNING: xxx operation is NOT available in the current drone position xxx: 操作名称	ドローンポジション(ホームポジションまたはホームポジション以外)に対して、無効な操作が行われています。
5	ERROR: "port_name" is NOT connected	Game Controller RTC とのデータポート未接続時に発生します。

9. FAQ

よくある質問を以下に示します。

表 9-1 FAQ 一覧

No.	Q：質問	A：回答
1	コントローラー操作が効きません	<p>以下の原因が考えられます。</p> <ul style="list-style-type: none">・ Game Controller が Active せず Game Controller のコンソールに赤文字でエラーメッセージが出ていますと、PS4 コントローラーが地上局と接続できていないことを示します。接続してもエラーが発生する場合は PS4 コントローラーの故障が考えられます。・ RTC のデータポート接続ミス 本書に沿って接続してください。・ PS4 コントローラーの電池不足 (=ライトバーが青色点灯せず) 充電が必要です。充電中はオレンジ色にてゆっくり点滅します。 有線接続でもフライト可能ですので、USB ケーブル (USB type-A<-->microUSB type-B) を別途ご用意ください。・ 最小高度、最大高度設定ミス 7.2 ドローンコントローラーRTC を参照し、適切な値を設定してください。

著作権

本文書の著作権は公立大学法人 会津大学に帰属します。

この文書のライセンスは以下のとおりです。

[クリエイティブ・コモンズ 表示 2.1 日本](http://creativecommons.org/licenses/by/2.1/jp/)

<http://creativecommons.org/licenses/by/2.1/jp/>

