

# 小型自動船 (Mini-V) を 作ってみた(作成中)

OUXT POLARIS

# 自己紹介：OUXT Polarisについて

## OUXT Polaris

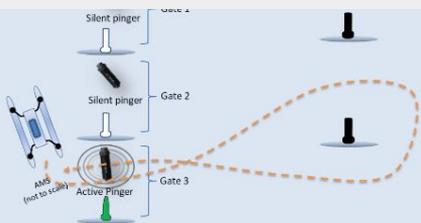
「Maritime RobotX Challenge」に挑戦する  
日本唯一のチーム。2014年大会から参加。  
社会人、学生の混成チーム

## Maritime RobotX Challenge

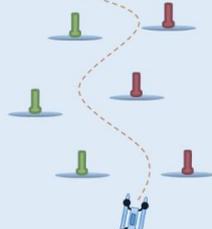
2年おきに開催される船の自動航行をテーマにした国際ロボットコンテスト  
(最近ではコロナで大会が中止になったり散々)  
次回は11月にシドニーで開催予定



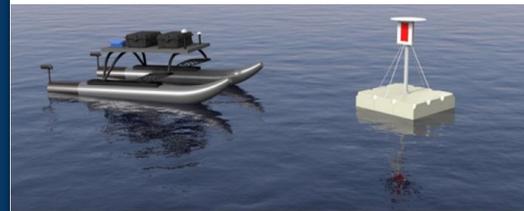
### 発振ピンガのゲートを通過



### ブイ緑/赤の間を自動航行



ブイの色によって次のタスクが変化  
(赤だったら右回りなど)



# 作った動機 コロナつらい

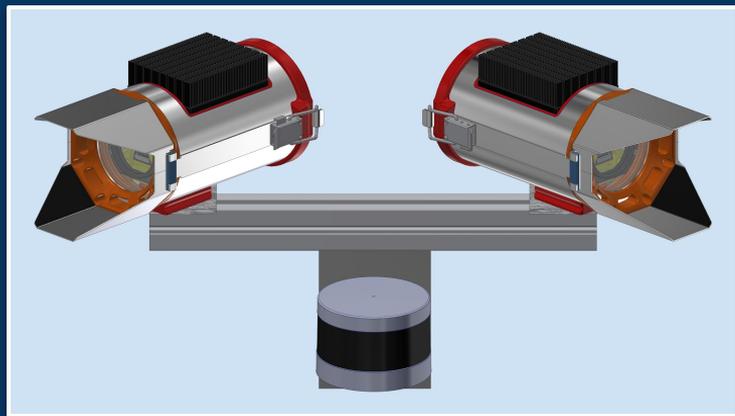
コロナつらい...つらくない?  
OUXTはつらいです.

対面で会ったり集まったり出来ない

- 船を見たことがないメンバーがいる
- 直接会ったことがないメンバーがたくさん
- 加工・組み立てが進まない/できない
- **実験が出来ない**

# 作った動機 コロナに抗うOUXT

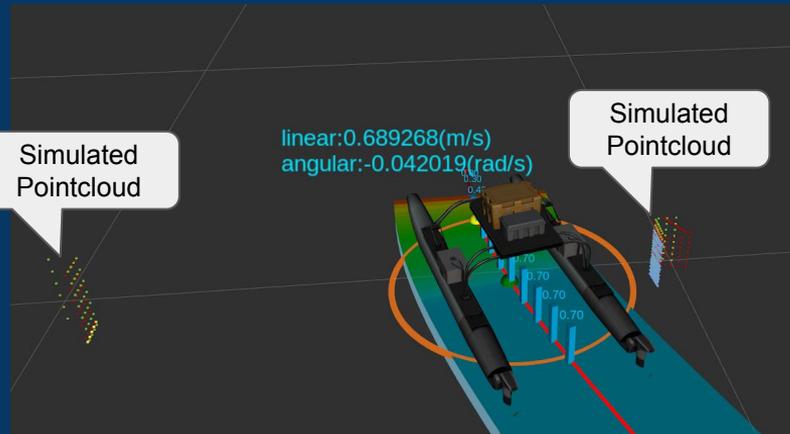
コロナ禍でも持続可能な開発を！



## センサアレイ

散らばったセンサを集めました

船がなくても組み立てできる  
これさえあれば1人でデータ  
収集が可能！



## シミュレータ

実機なしでも実験できる  
気軽に動作確認できる

# 作った動機 でもやっぱり実験したい！

究極の統合テスト：実機実験

でも大人数集まる大規模実験はちょっと....



小型船

船がデカイので規模がでかくなりがち



- 実機に近いセンサ構成
- 簡単に持ち運びできる
- 少人数で実験できる
- 個人宅で保管できる

# で、Mini-V作ってみた

( )は大会実機のサイズ/重量

1/5~10スケール程度の小型自動船を作成

- ・ 重量：13[kg](115kg)
- ・ ペイロード：5[kg]

## 運搬時の荷姿

キャリーケースで運搬可能



# 主機・電装品構成

まずはラジコン航行できる最小構成

計算機 (NUC) \*ハブの下

- ・ core i7
- ・ Ubuntu 20.04, ROS2



推進機

- ・ 防水用ブラシレス DCモータ

防水サーボ (近日搭載)

- ・ dynamixel XW540



バッテリー

- ・ 16Ahr
- \*電動自転車用

DC/DC x2

- \*秋月の安定化電源Kit

今後搭載

- ・ RTK-GPS
- ・ LTE無線
- etc

マイコン(搭載準備中)

- ・ NXP LPC1768

# アクチュエータ構成



防水  
Dynamixel  
ハイッテル(予  
定)



- アジマススラストx2
  - Dynamixel XW540-T260 x 2
  - U2D2からシリアル通信で制御
- 水中サーボ
  - F2838-350KV
  - LPC1768マイコンからPWM制御

しかし...まだ足りない...

# MINI-V 1.0の問題点

- 計算機構成が本番機と違う
  - 本番では二台構成の可能性もあるが、搭載容量的に... 🙄
- センサ構成も違う
  - 本番機のセンサを乗せると船が沈む！！
  - Hullが小さすぎてそこまで浮力が得られない 🙄
- 回路構成が本番機と違う
  - 本番機はMDが左右1機ずつ、1.0は左右で1機
  - これにより、MDとROSの通信プロトコルが本番機と異なる 🙄
  - そもそも回路デバッグ、ファームのデバッグができない

**本番と同じ回路  
積みたい！！**

# MINI-V 2.0の設計思想

そうだ、デカくしよう



カヌー用サイドフロートで大型化

大型化することで  
本番とおなじ計算  
機構成を実現

# メンバー募集

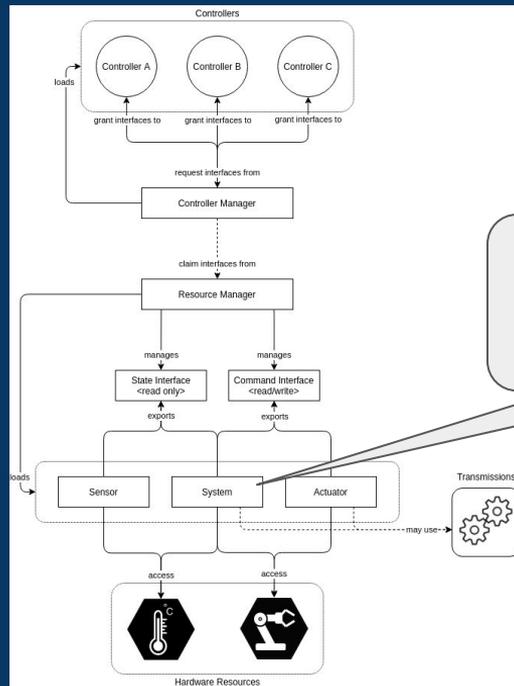
特に学生！

OUXT Polarisでは自律航行をやってみたい人を募集しています！

- ボールランチャー設計
- 電子回路設計
- ファームウェア実装
- カルマンフィルタによる自己位置推定
- 物体認識系の実装
- 運営（スポンサーとのやりとりなど）



上の内容等に興味のあるかたは  @OUXT\_Polaris まで！



URDFと制御パラメータ  
を書くだけで  
ROS2と一緒に  
Dynamixelが使える！！

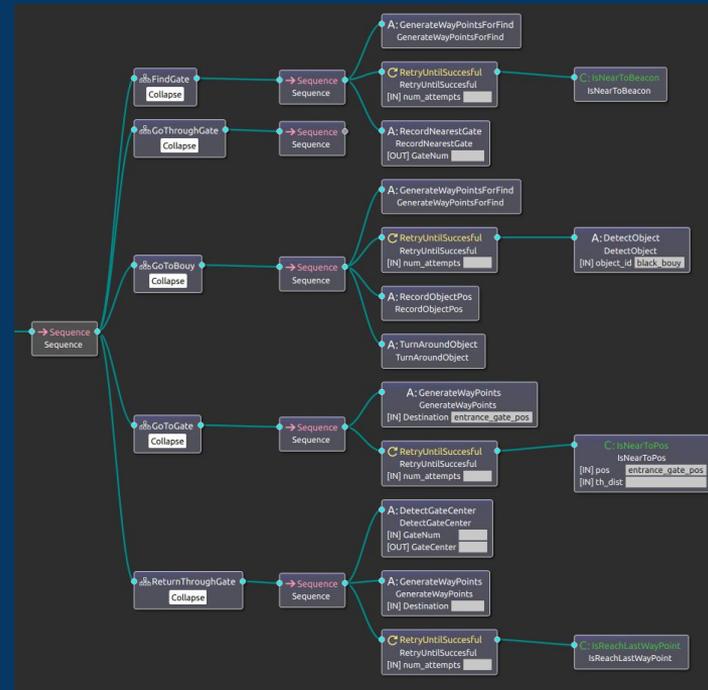
- 遠隔制御にはros2\_controlを使用
- ついついこだわって汎用Dynamixel Hardware Interfaceを作ってしまった
  - アドレステーブルを定義するクラスと基底型を継承したモータークラスを作れば任意のDynamixelに対応できる（はず）

# ソフトウェア(Mini-Vでも動かすものは一緒)

## タスクの記述 : BehaviorTree BehaviorTree.CPP + Groot

navigation2でも使われているよ

問題 : 組み方が難しい  
使いやすく組むには色々工夫が必要



アクション

引数

プログラム

例 : GoGoal() ⇒ Go(goal) (ノードと引数の境界移動)

# BehaviorTree.CPPの雰囲気とBlackBoard

## 設定ファイル

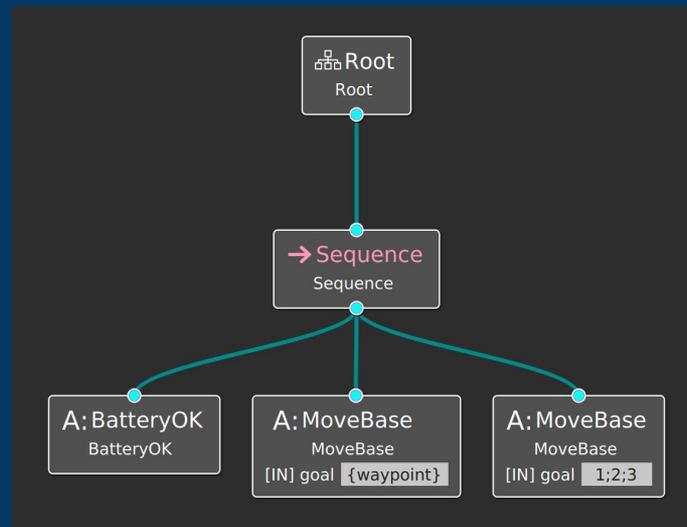
```
<root>
  <BehaviorTree>
    <Sequence>
      <BatteryOK/>
      <MoveBase
goal="{waypoint}"/>
      <MoveBase goal="1.2.3"/>
    </Sequence>
  </BehaviorTree>
</root>
```

BlackBoardから読み込み

## BlackBoard

設定ファイルの中の変数  
プログラムから設定できる  
(もちろん設定ファイルの中でも設定できる)

## Groot



# 歴史は繰り返す

## 設定ファイル

```
<root>
  <BehaviorTree>
    <Sequence>
      <BatteryOK/>
      <MoveBase
goal="{waypoint}"/>
      <MoveBase goal="1;2;3"/>
    </Sequence>
  </BehaviorTree>
</root>
```

四則演算したいね

関数とかも使えたら  
便利よね

アクション

引数

プログラム



そうだ、プログラマ  
ブルにしよう！



xacroが作られた  
時のようじゃ...

# プロトタイプ

## behavior:

### description :

"\$(find-pkg-share robotx\_setup)/task1.xml" ←ビヘイビア情報を記述した設定ファイル

### blackboard:

input : waypoint1

←書き換えたいBlackBoard

eval: "between(

filter(find(red,buoy)),

filter(find(green,buoy))"

←評価式(luaの書式が使える！)

## objects:

- uuid: 00000000

←事前知識

### attributes:

- red

- buoy

### pose:

#### position :

x: 0

y: 0

z: 0

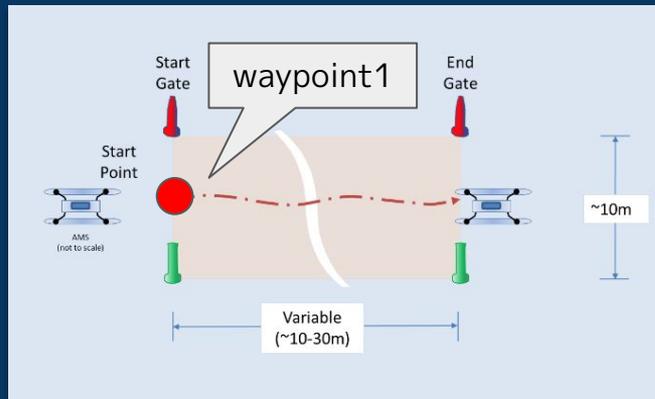
- uuid: 00000001

### attributes:

- green

- buoy

赤いブイと緑のブイの間を通過して！



OUXT勉強会を再開します！

次回 5/14(金) 22:00~

シミュレータについて (片岡大哉)

<https://ouxt-polaris.connpass.com/event/210684/>

## 今後の勉強会の予定

- センサアレイについて
- MiniVのハードウェアドライバについて
- 船の制御について
- ビヘイビアについて(今日の続きとか)



※聴講料は社会人の方に関しましては今後有料化する予定です